

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

Colour Consistency in Computer Vision: A Multiple Image Dynamic Exposure Colour Classification System

A THESIS PRESENTED TO THE
INSTITUTE OF NATURAL AND MATHEMATICAL SCIENCES
IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN COMPUTER SCIENCE
AT
MASSEY UNIVERSITY, ALBANY,
AUCKLAND,
NEW ZEALAND

BY
SAM MCGHIE
FEBRUARY 2016

Abstract

Colour classification vision systems face difficulty when a scene contains both very bright and dark regions. An indistinguishable colour at one exposure may be distinguishable at another. The use of multiple cameras with varying levels of sensitivity is explored in this thesis, aiding the classification of colours in scenes with high illumination ranges. Titled the Multiple Image Dynamic Exposure Colour Classification (MIDECC) System, pie-slice classifiers are optimised for normalised red/green and cyan/magenta colour spaces. The MIDECC system finds a limited section of hyperspace for each classifier, resulting in a process which requires minimal manual input with the ability to filter background samples without specialised training. In experimental implementation, automatic multiple-camera exposure, data sampling, training and colour space evaluation to recognise 8 target colours across 14 different lighting scenarios is processed in approximately 30 seconds. The system provides computationally effective training and classification, outputting an overall true positive score of 92.4% with an illumination range between bright and dim regions of 880 lux. False positive classifications are minimised to 4.24%, assisted by heuristic background filtering. The limited search space classifiers and layout of the colour spaces ensures the MIDECC system is less likely to classify dissimilar colours, requiring a certain ‘confidence’ level before a match is outputted. Unfortunately the system struggles to classify colours under extremely bright illumination due to the simplistic classification building technique. Results are compared to the common machine learning algorithms Naïve Bayes, Neural Networks, Random Tree and C4.5 Tree Classifiers. These algorithms return greater than 98.5% true positives and less than 1.53% false positives, with Random Tree and Naïve Bayes providing the best and worst comparable algorithms, respectively. Although resulting in a lower classification rate, the MIDECC system trains with minimal user input, ignores background and untrained samples when classifying and trains faster than most of the studied machine learning algorithms.

Acknowledgements

I would like to thank my supervisors Dr. Napoleon Reyes and Dr. Andre Barczak, without their collective input and guidance I wouldn't finish this thesis. Countless hours of discussions and colourful whiteboard diagrams have guided my research, expanded my thought process and enabled me to view my research from different angles. Their enthusiasm and knowledge has piqued my interest to further study in the field.

Contents

Abstract	iii
Acknowledgements	v
Contents	vii
List of Tables	xi
List of Figures	xiv
List of Algorithms	xviii
1. Research Description	1
1.1 Overview of the Current State of Technology	1
1.2 Research Objectives.....	2
1.2.1 General Objective	2
1.2.2 Specific Objectives	2
1.3 Significance of Research	3
1.4 Scope and Limitations of Research	3
1.5 Structure of Thesis	4
2. Theoretical Framework	5
2.1 Colour	5
2.2 Colour Spaces	6
2.2.1 HSL and HSV colour models	7
2.2.2 Normalised Red/Green Chromaticity	7
2.3 Image Exposure	8
2.4 Homographic Transformations	9
2.5 Fuzzy Logic	10
2.6 Background to Comparison Classifiers	11
2.6.1 Naïve Bayes Classifier.....	11
2.6.2 J48 Tree Classifier	12

2.6.3	Random Tree Classifier	13
2.6.4	Neural Network Classifier	13
2.7	Summary	15
3.	Review of Related Literature	17
3.1	Colour Classification	17
3.1.1	Robustness of Colour Detection for Robot Soccer	17
3.1.2	Knowledge-Based Fuzzy Colour Processing	19
3.1.3	Colour Classification using margin-setting with ellipsoids	22
3.2	Contrast Adjustment	24
3.2.1	Colour image enhancement using correlated intensity and saturation adjustments	24
3.2.2	Dynamic Colour Object Recognition Using Fuzzy Logic	26
3.3	Image Exposure Selection	30
3.3.1	Acquisition of Agronomic Images with Sufficient Quality by Automatic Exposure Time Control and Histogram Matching	30
3.3.2	OCR Accuracy Improvement on Document Images through a novel pre-processing approach	33
3.4	Multiple Camera Input and Processing	37
3.4.1	A unified framework for multi-sensor HDR video reconstruction	37
3.4.2	Object Tracking in the Presence of Occlusions Using Multiple Cameras: A Sensor Network Approach	38
3.4.3	3D Scene Analysis by Real-Time Stereovision	40
3.5	Summary	42
4.	Multiple Image Dynamic Exposure Colour Classification (MIDECC) System	47
4.1	MIDECC General Architecture	48
4.2	Introducing the Normalised Cyan/Magenta Colour Space	50
4.3	Multiple Camera Input Pre-Processing	51
4.3.1	Camera Calibration	52
4.3.2	Scenario and Lighting Capture	53
4.3.3	Exposure Selection Process	54
4.4	Sampling Training and Test Data	56
4.4.1	Conversion to Pie Slice	57

4.4.2	Data Collection and Storage	57
4.5	Building Classifiers.....	59
4.5.1	Pie Slice Classifier Construction	59
4.5.2	Colour Space Selection Process.....	60
4.5.3	Standard Deviation Priority during Classification.....	61
4.5.4	Handling Multiple Classifications	63
4.6	Background Removal	64
4.7	Evaluating Classifiers	64
4.7.1	Classifier Evaluation Methods.....	65
4.7.2	Sampling Test Data.....	65
4.8	Summary.....	66
5.	Experimental Setup	67
5.1	Environment Setup	67
5.1.1	Cameras.....	67
5.1.1.1	Locations.....	68
5.1.1.2	Transformations	68
5.1.2	Known-Area Colour Patches	70
5.1.2.1	Colour Choices	71
5.1.2.2	Patch Locations.....	71
5.1.2.3	Manual Colour Entry	72
5.1.3	Lighting.....	72
5.1.3.1	Scenarios.....	73
5.2	Exposure Selection	75
5.3	Lighting Scenario Capture	76
5.4	Building Classifiers.....	76
5.5	Classifier Evaluation.....	80
5.6	Background Removal	82
6.	Experiment Results and Analysis	85
6.1	System Results; Known-Areas Only	85
6.1.1	Naïve Bayes Classifiers	92
6.1.2	J48 Tree Classifiers.....	93
6.1.3	Random Tree Classifiers.....	94

6.1.4 Neural Network Classifiers.....	95
6.2 System Results; Known-Areas and Background.....	97
6.2.1 Naïve Bayes Classifiers	104
6.2.2 J48 Tree Classifier	105
6.2.3 Random Tree Classifier	106
6.2.4 Neural Network Classifier	107
6.3 Summary.....	108
6.4 Discussion.....	109
6.4.1 Harsh Lighting Limitations.....	109
6.4.2 Classification with FCCF.....	112
7. Conclusions	115
7.1 Suggestions for Future Work.....	116
Bibliography	117
Appendix A: Experiment Implementation	119
Appendix B: Visual MIDDEC System Results	121

List of Tables

Table 4-1: Training and test set class sizes.....	58
Table 4-2: Example of CSV file layout for WEKA Classifier training and testing	58
Table 4-3: Data Collected for Training and Testing.....	65
Table 5-1: Camera Alignment values	69
Table 5-2: Colours sampled from pre-processed images	71
Table 5-3: Lighting Scenarios	74
Table 5-4: Locked exposure settings for each camera.....	75
Table 5-5: Colour Space selection example for Camera A, Red.....	78
Table 5-6: Colour Space selection example for Camera A, Light Green.....	78
Table 5-7: Colour Space selection example for Camera B, Red.....	79
Table 5-8: Colour Space selection example for Camera B, Light Green	79
Table 5-9: Final Classifiers to be used by the MIDECC System	80
Table 6-1: Training sample sizes, known-areas only	85
Table 6-2: Classifier training set results, known-areas only.....	86
Table 6-3: MIDECCS training set results, per scenario, known-areas only.....	86
Table 6-4: Classifier test set results, known-areas only	87
Table 6-5: MIDECCS test set results, per scenario, known areas only	87
Table 6-6: MIDECCS Confusion matrix for 'Even Bright' training scenario, known-areas only	88
Table 6-7: MIDECCS Confusion matrix for 'Uneven Moderate 2' test scenario, known-areas only	89
Table 6-8: MIDECCS Confusion matrix for 'Harsh Moderate 1' test scenario, known-areas only	90
Table 6-9: MIDECCS Confusion matrix for all 14 test scenarios, known-areas only	91
Table 6-10: MIDECCS Confusion matrix for all 14 test scenarios, known-areas only	91
Table 6-11: Naïve Bayes Confusion matrix for all 14 training scenarios, known-areas only	92
Table 6-12: Naive Bays Confusion matrix for all 14 test scenarios, known-areas only	92
Table 6-13: J48 Confusion matrix for all 14 training scenarios, known-areas only.....	93

Table 6-14: J48 Confusion matrix for all 14 test scenarios, known-areas only	93
Table 6-15: Random Tree Confusion matrix for all 14 training scenarios, known-areas only.....	94
Table 6-16: Random Tree Confusion matrix for all 14 test scenarios, known-areas only.....	94
Table 6-17: Neural Networks Confusion matrix for all 14 training scenarios, known-areas only.....	95
Table 6-18: Neural Networks Confusion matrix for all 14 test scenarios, known-areas only.....	95
Table 6-19: Training sample sizes, including background	97
Table 6-20: Classifier training set results, including background.....	98
Table 6-21: MIDECCS training set results, per scenario, including background.....	98
Table 6-22: Classifier test set results, including background.....	99
Table 6-23: MIDECCS test set results, per scenario, including background.....	99
Table 6-24: MIDECCS Confusion matrix for 'Even Bright' training scenario, including background	100
Table 6-25: MIDECCS Confusion matrix for Uneven Moderate 2' test scenario, including background	101
Table 6-26: MIDECCS Confusion matrix for 'Harsh Moderate 1' test scenario, including background	102
Table 6-27: MIDECCS Confusion matrix for all 14 training scenarios, including background	103
Table 6-28: MIDECCS Confusion matrix for all 14 test scenarios, including background	103
Table 6-29: Naive Bays Confusion matrix for all 14 training scenarios, including background	104
Table 6-30: Naive Bays Confusion matrix for all 14 test scenarios, including background	104
Table 6-31: J48 Confusion matrix for all 14 training scenarios, including background	105
Table 6-32: J48 Confusion matrix for all 14 test scenarios, including background	105
Table 6-33: Random Tree Confusion matrix for all 14 training scenarios, including background	106
Table 6-34: Random Tree Confusion matrix for all 14 test scenarios, including background	106

Table 6-35: Neural Network Confusion matrix for all 14 training scenarios, including background.....	107
Table 6-36: Neural Network Confusion matrix for all 14 test scenarios, including background.....	107
Table 6-37: FCCF Rules derived using brute force, as outlined in [1].....	112

List of Figures

Please note that unless otherwise specified, Figures representing inputs or outputs of the MIDECC System are positioned in one of the two following ways:

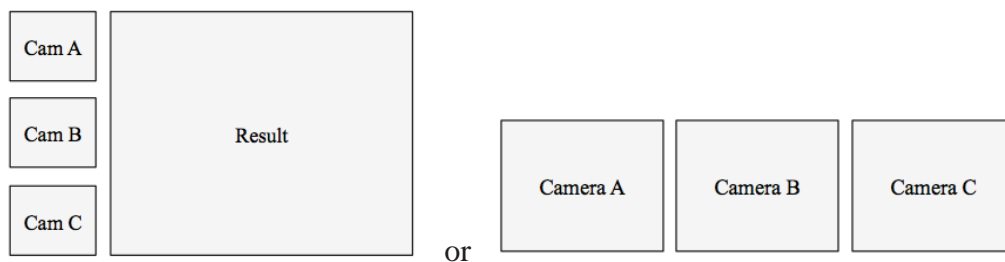


Figure 2-1: Electromagnetic Spectrum with the visible spectrum highlighted	5
Figure 2-2: Bayer Colour Filter Array (left), Photosites with Colour Filters (right)	6
Figure 2-3: Example of a RGB colour cube.....	6
Figure 2-4: HSL Cylinder (left), HSV Cylinder (right)	7
Figure 2-5: rg-chromaticity (left)	8
Figure 2-6: Differences between an under, over and correctly exposed image	8
Figure 2-7: Diagram representing a simple homographic adjustment	9
Figure 2-8: Fuzzy Logic membership function.....	10
Figure 2-9: Single Layer Neural Network.....	14
Figure 3-1: Rectangular decision region in the YUV colour space	18
Figure 3-2: Pie Slice classifier in the YUV colour space.....	18
Figure 3-3: ROI for a Resistance Spot Weld (left), Table depicting Quality Criteria for comparing good and poor welding spots (right).....	19
Figure 3-4: HSI Model for $I = 1.0$	21
Figure 3-5: Fuzzy set, defined over the HS-colour space (left), Definition of the colour red (right)	21
Figure 3-6: Frequency distribution of a good sample (left) compared to a poor sample (right).....	22
Figure 3-7: Result of filtering the source image into class one (left) and class two (right) with a zero-margin classifier	23

Figure 3-8: Result of filtering the source image into class one (left) and class two (right) with a 25% margin classifier.....	23
Figure 3-9: ISAISC General System Architecture	25
Figure 3-10: Source Image (top), ISAIS Result (left), ISAISC Result (right)	26
Figure 3-11: Pie Slice colour decision region in rg-colour space (left), rg-colour space (right).....	27
Figure 3-12: Contrast intensification operator.....	28
Figure 3-13: Contrast degradation operator.....	29
Figure 3-14: General Architecture of the Automatic ET Control and Histogram Matching System.....	30
Figure 3-15: Input Image highlighting the ROI (left), Histogram for the RGB channels in the ROI (right).....	31
Figure 3-16: Images captured with different exposure times for adjusting parameters	32
Figure 3-17: Identification of crop lines and weed patches in the ROI.....	32
Figure 3-18: Block Diagram of the OCR pre-processing approach	33
Figure 3-19: Original Image (left), Enhanced image using CLAHE equalisation (right).....	34
Figure 3-20: Original Image (left), Enhanced image using the proposed brightness and contrast adjustments (right)	35
Figure 3-21: Image after contrast and brightness adjustment (left), Result of luminance algorithm for grey scale conversion (right)	35
Figure 3-22: Grey scale image (left), Result of un-sharp masking filter (right).....	35
Figure 3-23: Grey scale sharpened image (left), Result of binarised image using Otsu thresholding approach (right).....	36
Figure 3-24: Common HDR System processing steps	37
Figure 3-25: Illustration of the Object Tracking in the Presence of Occlusions problem setup.....	39
Figure 3-26: Perspective Camera measurement model, with unoccluded object at x	39
Figure 3-27: 3D Scene Analysis System General Architecture.....	41
Figure 3-28: Example of Epipolar Geometry	41
Figure 4-1: MIDECC System Classification Architecture	48
Figure 4-2: MIDECC System Training Architecture	49
Figure 4-3: Normalised R/G Colour Space (left) and Normalised C/M Colour Space (right) with areas of strength highlighted	50
Figure 4-4: Exposure ranges spread across three input cameras	51
Figure 4-5: Examples of perspective-skewed images from each input camera.....	52

Figure 4-6: Even Lighting Scenario examples	53
Figure 4-7: Uneven Lighting Scenario examples.....	53
Figure 4-8: Harsh Lighting Scenario examples.....	53
Figure 4-9: MIDECCS Exposure Selection Process	54
Figure 4-10: Incorrect sampling of known-areas, without margins	56
Figure 4-11: Correct sampling of known-areas, with margins.....	57
Figure 4-12: Pie Slice classifier, indicating the two priority regions	59
Figure 4-13: Example of multiple classifiers matching Dark Blue.....	62
Figure 4-14: Example of multiple classifiers matching Yellow	62
Figure 4-15: Example of multiple classifiers matching, with no priority	63
Figure 4-16: Grey Area, highlighted in the R/G colour space	64
Figure 5-1: Logitech Camera locations, emphasis added	68
Figure 5-2: Camera Raw Input images with red outline around focus plane (top), same images with pre-processing completed (bottom)	69
Figure 5-3: Camera Alignment testing output.....	70
Figure 5-4: Sample of known-area colour patches on the focusing plane	70
Figure 5-5: Area 2 highlighting the importance of the margin value for each camera	71
Figure 5-6: Patch locations throughout the focusing plane, viewed through multiple cameras.....	71
Figure 5-7: Manual colour entry samples (left), the manually classified pie slice (centre), with tolerance adjustments (right)	72
Figure 5-8: Room lighting locations and terminology	73
Figure 5-9: Measuring illumination in the centre of the plane (left) and in corners (centre), corner terminology (right).....	74
Figure 5-10: Multiple camera inputs highlighting exposure spread for scenario 'Harsh Severe 1'	75
Figure 5-11: MIDECCS Experimental Program outputting classifiers in both colour spaces with corresponding statistics.....	76
Figure 5-12: Grey Area filtering disabled, producing a wider pie slice classifier (left), Grey Area filtering enabled at 0.015, producing the preferred classifier (right).....	77
Figure 5-13: Classifier Layout for Camera A	81
Figure 5-14: Classifier Layout for Camera B.....	81
Figure 5-15: Classifier Layout for Camera C.....	81
Figure 5-16: MIDECCS Output of 'Even Bright' training scenario with background removal turned off (left half of image) compared to turned on (right half of image)	82

Figure 5-17: MIDECCS Output for 'Even Bright' training scenario with statistic shading	83
Figure 5-18: Chart indicating minimal difference in true positives with background removal.....	84
Figure 6-1: Inputs (left) and MIDECCS result (right) of 'Even Bright' training scenario, known-areas only	88
Figure 6-2: Inputs (left) and MIDECCS result (right) of 'Uneven Moderate 2' test scenario, known-areas only	89
Figure 6-3: Inputs (left) and MIDECCS result (right) of 'Harsh Moderate 1' test scenario, known-areas only	90
Figure 6-4: Single Layer Neural Network, generated and trained by WEKA.....	96
Figure 6-5: Inputs (left) and MIDECCS result (right) of 'Even Bright' training scenario, including background.....	100
Figure 6-6: Inputs (left) and MIDECCS result (right) of 'Uneven Moderate 2' test scenario, including background.....	101
Figure 6-7: Inputs (left) and MIDECCS result (right) of 'Harsh Moderate 1' test scenario, including background.....	102
Figure 6-8: Example of sample data for Red.....	110
Figure 6-9: Classifier generated for 1x Standard Deviation either side of the mean	110
Figure 6-10: Extended search space, generated by 2x Standard Deviations either side of the mean	110
Figure 6-11: Clockwise, from upper left: 1x Standard Deviation, 2x Standard Deviations, 3x Standard Deviations, 4x Standard Deviations	111

List of Algorithms

Algorithm 2-1: C4.5 Classification 12

Algorithm 2-2: Neural Network Back Propagation 14

Algorithm 4-1: Camera A Exposure Selection
 (Best Overall) 55

Algorithm 4-2: Camera B Exposure Selection
 (Darker Exposure, Focuses on Brighter Areas)..... 55

Algorithm 4-3: Camera C Exposure Selection
 (Brighter Exposure, Focuses on Darker Areas)..... 55

Algorithm 4-4: Matching a classifier with multiple priority levels 61

Chapter 1

1. Research Description

Reliably classifying colours in a computer vision system is a difficult and often computationally expensive task. In comparison, the human vision system is well apt at both detecting subtle differences in similar colours and recognising the same colour when subject to varying illuminations.

1.1 Overview of the Current State of Technology

Current implementations of colour classifiers, such as Fuzzy Colour Contrast Fusion (FCCF) successfully compensate for limited illumination variation, searching a pie-slice decision region using colour contrast rules to strengthen or weaken matches to a particular colour classification [1]. FCCF requires the user to manually calibrate each colour into a pie-slice decision region, before using a brute-force method to choose contrast operations to run on the incoming red, green and blue channels. It will be imperative to this research to automate as much of the classifier generation as possible, while assessing if FCCF is a valid addition to the processing pipeline.

A current implementation of automatic exposure practices histogram matching to compare a stored reference value with the current input image [2]. While this method is robust enough to be run in real time, this research will focus on the use of no external reference points, minimising human interaction once the training process has begun.

Other research has been conducted into which the background of a scanned text document is removed, emphasising the typescript and images later to be scanned by an Optical Character Recognition (OCR) program [3]. The various steps to the overall process and use of a standardised ‘Ostu’s method’ of binary thresholding will be amongst the techniques used to reduce processing time for this research paper.

Different approaches have been reviewed regarding multiple camera inputs, with one paper suggesting the distribution of computational processing throughout the network, reducing camera system traffic [4]. Reducing the amount of processing by only running the minimum required on a ‘slave’ camera is also studied [5], while another paper highlighting the difficulties in multiple camera alignment and processing [6].

1.2 Research Objectives

1.2.1 General Objective

To develop a robust and efficient automatic colour classification system which assesses multiple frames taken simultaneously from three different cameras. Each camera is to have a distinct exposure setting, widening the dynamic range captured.

The system will endeavour to find an optimal classifier, providing robust results when classifying colours in as many bright and dim lighting scenarios as possible.

1.2.2 Specific Objectives

1. To investigate how three input images may be combined to provide select regions where each specialises in a particular exposure range for the explicit task of colour recognition.
2. To investigate the reduction of ambiguity by prioritising matches when multiple classifiers do not return a unanimous result.
3. To investigate the use of a novel colour space, ‘normalised cyan/magenta’ to complement the strengths of the normalised red/green colour space using a pie-slice classifier.
4. To investigate background image removal without reducing the accuracy of colour classification.
5. To investigate the feasibility of implementing FCCF to strengthen classification results.

1.3 Significance of Research

This research employs multiple sub-processes to create a simple multiple image colour classification system:

- A unique colour space is introduced, complementary to previous research to aid the classification in difficult illumination (Section 4.2)
- An automatic exposure process calibrates each camera exposure, resulting in each camera specialising in a particular brightness region (Section 4.3.3)
- Colour classifiers are spread amongst two colour spaces, extending the feature classification range while using only the ‘preferred’ space, reducing processing time (Section 4.5.2)
- A simple, yet effective method of prioritising classifiers is used if multiple classifiers match, providing reliable output (Section 4.5.4)

The MIDECC system trains in a supervised learning environment, faster than three of the four common machine learning algorithms discussed in Section 2.6. Once the data is collected for each colour, assuming a normal distribution quickly calculates a pie slice classifier, discussed in Section 4.5.1.

1.4 Scope and Limitations of Research

This research is limited to the colours definable in the normalised red/green colour space. Due to the layout of the colour space, this excludes colours such as white, grey and black.

It is assumed that the colour patches are a matte finish and are an acceptable size to be recognised by the camera at a reasonable distance. It is also assumed that the lighting of the testing environment is relatively free of colour cast or shadows that may disrupt the training process.

Much of the proposed system will be automated, however some values may require manual fine-tuning. Manual parameters to be provided are the sampling of colours for the exposure selection process (Section 4.3.3) and the threshold at which colours are classified as background pixels (Section 4.6). Setting these parameters do not require specialist knowledge of the system, as settings will be easily accessible and their actions immediately observable in the system output.

The system will be tested to classify illumination varying from 175 lux to 11,230 lux. It is expected that the system may struggle at these extremities, however limitations and evaluated ranges will be discussed, concluding this thesis.

1.5 Structure of Thesis

This thesis begins in Chapter 2 by covering concepts such as colour, image adjustments such as contrast and sharpness, geometric transforms and the use of Fuzzy Logic. Introductions to the comparison classifiers are given to provide a background to the standardised testing the system is evaluated against in Section 2.6.

Previous research is discussed in Chapter 3, exploring papers which focus on colour classification, image contrast adjustment, image exposure selection and multiple image processing.

A complementary colour space, the Normalised Cyan/Magenta chromaticity is introduced in Section 4.2, along with the system process for exposure selection, colour classification, and evaluation in the remainder of Chapter 4. These processes combine, creating the MIDECC system, implemented Chapter 5. Results will be discussed in Chapter 6 and conclusions drawn in Chapter 7.

Chapter 2

2. Theoretical Framework

2.1 Colour

Observed by H. E. Smithson, the human sensory system is apt at distinguishing an objects' colour constantly despite changes in illumination [7]. The ability to judge the spectral reflectance or colour, despite changing illumination is described by the term 'colour consistency'.

The human visual system senses colour based on three spectral receptive cones, namely the S-, M- and L-cones. Known as a trichromatic system, colour information is conveyed as three separate channels to these cones as blue, green and red. Photoreceptors known as rods detect brightness. The colour of an object is perceived as a combination of reflected and absorbed wavelengths in the visible spectrum.

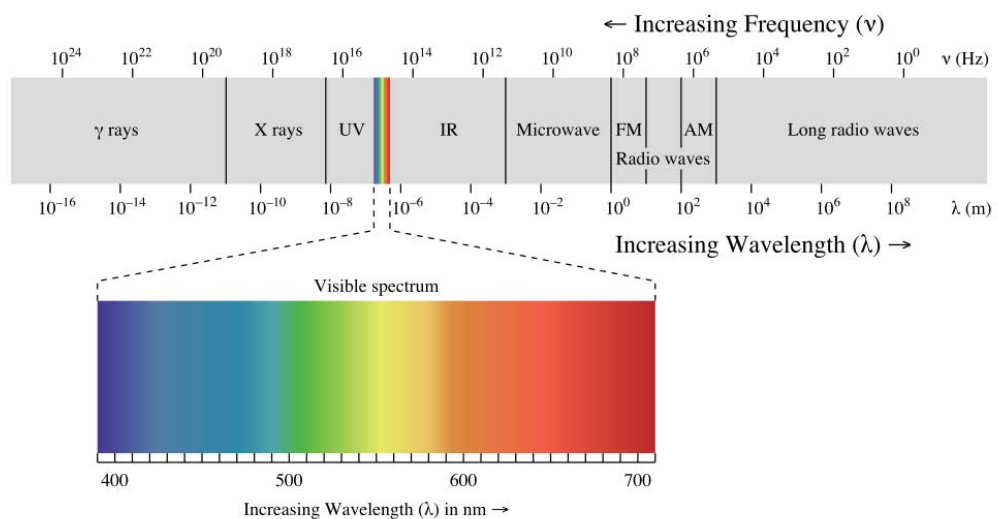


Figure 2-1: Electromagnetic Spectrum with the visible spectrum highlighted

Figure created by Philip Ronan, Wikipedia

Digital camera sensors record the visible wavelengths of light by using an array of small ‘photosites’, assessing how many photons of light falls into each in a given exposure time [8]. The photosites are organised into an array, with colour filters covering the sensors in a certain pattern. The pattern below is the most common used in digital cameras, the Bayer array:

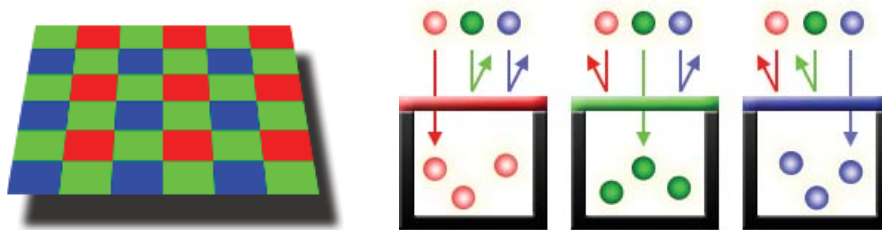


Figure 2-2: Bayer Colour Filter Array (left), Photosites with Colour Filters (right)

This pattern results in twice as many green sensors compared to blue and red. This is due to the human eye being more sensitive to green light, as producing an image with more green light appears less noisy and highlights greater detail when seen by the human eye.

2.2 Colour Spaces

A colour space is an organised set of colours that allow reproductions of colour both using digital and analogue representations.

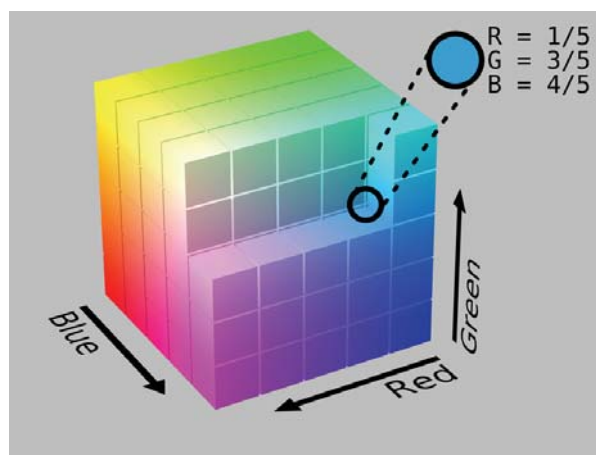


Figure 2-3: Example of a RGB colour cube

Figure created by Michael Horvath, Wikipedia

Figure 2-3 is representative of the simple RGB colour space, where each colour is represented by Red, Green and Blue ratios.

2.2.1 HSL and HSV colour models

The Hue, Saturation and Lightness (HSL) and the Hue, Saturation and Value (HSV) models were developed in the 1970s for computer graphics applications. While the tertiary attribute ‘Lightness’ and ‘Value’ differs greatly, the Hue and Saturation values are identical.

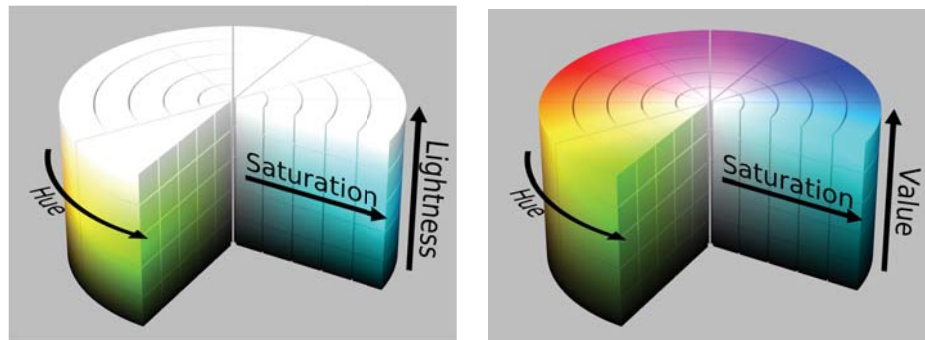


Figure 2-4: HSL Cylinder (left), HSV Cylinder (right)

Figures created by Michael Horvath, Wikipedia

Existing in the RGB colour space, the HSI model is often used in Computer Vision applications. Colour images in RGB format are converted to HSI, as assessing Red, Green and Blue values of an object relates to the amount of light hitting said object, compared to an actual hue. The values of Hue and Saturation provide a computer vision system with more information about the reflected wavelengths, whereas the Intensity value indicates the radiance of reflected light only.

2.2.2 Normalised Red/Green Chromaticity

The normalised red/green colour space or ‘rg chromaticity’ is created by normalising the red and green channels of the RGB colour space. The blue channel may be omitted as once normalised, the red and green channels both provide proportions of the resulting colour, summing to one. This means the blue value may be calculated if required.

The rg chromaticity does not include intensity information, therefore it is ideal for identifying colours where changes in illumination may negatively effect other colour models [1].

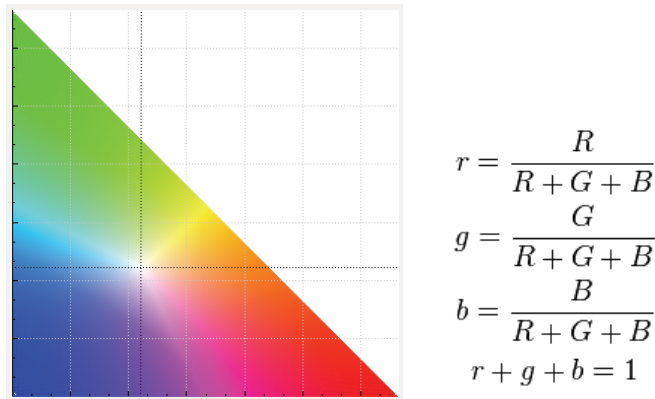


Figure 2-5: rg-chromaticity (left)

The above formulae shows the conversion from the intensity of Red (R), Green (G) and Blue (B) in an original colour being converted to normalised red (r), green (g) and blue (b) values. This colour model excels with a pie-slice classifier, moving the origin to (0.333, 0.333), indicated in Figure 2-5.

2.3 Image Exposure

The photographic term of exposure refers to the time a camera shutter is open in order to capture an image. An image is described as ‘over exposed’ when areas of interest are ‘washed out’, or seen as solid white. In contrast, an ‘under exposed image’ is when areas of interest are ‘muddy’ or too dark to distinguish [9].

A faster exposure time results in the shutter of a camera being open for less time, capturing a darker image with less blur. An image taken with a faster exposure time will result in more detail in areas with a greater illumination. Increasing the exposure time will produce a brighter image, capturing more detail in areas with less illumination.



Figure 2-6: Differences between an under, over and correctly exposed image
Image from Exposure Guide (<http://www.exposureguide.com/exposure.htm>)

2.4 Homographic Transformations

Homographic processing is to be used extensively in this thesis, removing any rotation or perspective skewness due to the camera position. Homography is frequently used in computer vision applications, introduced to study the appearance of objects from two different points of view.

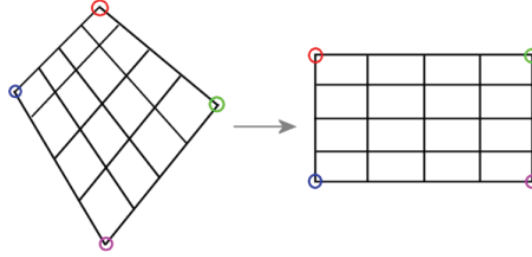


Figure 2-7: Diagram representing a simple homographic adjustment

Figure created by Xavier Philippeau, Developpez.com

Seen above, each of the four vertices of the input grid on the left side induces a bijection through a homographic relationship, straightening the image.

The OpenCV C++ programming library is to be used to calculate the perspective transformations for this thesis, specifically using the `findHomography()` function. While this thesis employs the use of an optimised function, this section explains the key equations used in this important step of image preparation. This function returns the perspective transformation H between the source and destination planes:

$$s_i \begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} \sim H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

So that the back-projection error, below, is minimised:

$$\sum_i \left(x'_i - \frac{h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + h_{33}} \right)^2 + \left(y'_i - \frac{h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + h_{33}} \right)^2$$

The resulting matrix returned from the OpenCV function may then be sent to the `warpPerspective()` function, which applies the transform to the image [10], [11].

2.5 Fuzzy Logic

A form of many-valued logic, fuzzy logic uses linguistic rules to apply weighted values to an output. A simple example would be a single input, single output system controlling the speed of a cooling fan.

IF temperature **IS** cold **THEN** *stop fan*
IF temperature **IS** warm **THEN** *slow fan*
IF temperature **IS** hot **THEN** *fast fan*

As seen by the rules, the input is a temperature value, while the output is the fan speed. Firstly, the temperature must be classified as one of the three values - cold, warm or hot. The outputs are to be a specified value, for example 'stop fan' may be stationary, 'slow fan' may be 600 revolutions per minute (rpm), while 'fast fan' may be 1350 rpm. This is matched by the use of a membership function, outputting a weighted value for each input classifier.

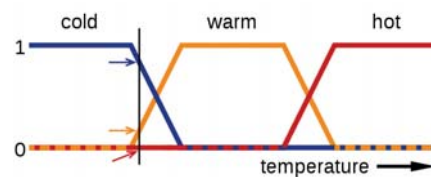


Figure 2-8: Fuzzy Logic membership function
Figure created by Wikipedia user 'fullofstars'

The measured temperature is indicated along the X-axis, while the Y-axis represents the weighting of each rule to be evaluated. The black line, indicated by the three arrows in Figure 2-8, shows a moderately cold input temperature. It is estimated that the weightings of each rule would be as follows:

(80%) IF temperature **IS** cold **THEN** *stop fan*
(20%) IF temperature **IS** warm **THEN** *slow fan*
(0%) IF temperature **IS** hot **THEN** *fast fan*

This would result in the system combining the outputs of 'stop' (80% of 0 rpm) with 'warm' (20% of 600rpm). This would output a final 'defuzzified' value of 120rpm.

While it is unlikely the membership functions of this research will be overlapping, the use of linguistic terms will be used to aid processing.

2.6 Background to Comparison Classifiers

The following classifiers are to be compared to the MIDECC system, as a method of judging the systems' training time and overall accuracy. The classifiers are trained and evaluated using the Waikato Environment for Knowledge Analysis (WEKA), Java-based open source software developed by the University of Waikato [12].

2.6.1 Naïve Bayes Classifier

The Naïve Bayes technique is a simple process, assigning 'class labels' to problem values. The technique then assumes that for each class label, the value of a particular feature is independent to any other feature. Although very simplistic, Naïve Bayes classifiers have functioned well when compared to other, more complex classifiers [13]. Bayesian classifiers, such as the Naïve Bayes classifiers, follow Bayes Theorem [14], below:

$$p(c_j|d) = \frac{p(d|c_j) p(c_j)}{p(d)}$$

Where

$p(c_j|d)$ = Probability of instance d being in class c_j

What is to be computed

$p(d|c_j)$ = probability of generating instance d given class c_j

Classified in class c_j , the probability of having feature d

$p(d)$ = probability of instance d occurring

Constant value for all classes

The Bayes Theorem, when used with a Naïve Bayes classifier, assumes that all attributes have independent distributions, estimating the following:

$$p(c_j|d) = p(d_1|c_j) * p(d_2|c_j) * ... * p(d_n|c_j)$$

By assuming independent distributions, Naive Bayes is both storage and computationally efficient. The classifier is often seen as a 'baseline' for category sorting algorithms such as spam filters, being able to operate on more complex data sets in a supervised learning environment, where training data has the desired output or supervisory signal.

2.6.2 J48 Tree Classifier

A Java implementation of the C4.5 algorithm developed by Ross Quinlan, this classifier generates a decision tree, often referred to as a statistical classifier [15]. The decision tree operates on the general algorithm below, splitting the data at the attribute with the highest normalisation gain.

The algorithm attempts to match the following base cases first:

- All the samples in the list belong to the same class.
The end of the branch - the final output value has been found.
- None of the features provide any information gain.
The end of the branch for half the equation, continue for the remaining data.
- Instance of previously-unseen class encountered.
The end of the branch for half the equation, continue for the remaining data.

If these cases are not met, the following algorithm is run:

Algorithm 2-1: C4.5 Classification

For each attribute *a*

Find the normalised information gain ratio from splitting *a*

Let *a_best* be the attribute with the highest normalised information gain

Create a decision *node* that splits at *a_best*

Recur on the sub-data at *a_best*, adding nodes as the children of *node*

The C4.5 algorithm and J48 implementation became popular after ranking best in the ‘Top 10 Algorithms in Data Mining’ paper, published in 2008 [16].

When used in this research, WEKA is instructed to generate a J48 decision tree with a pruning confidence threshold of 0.25 and a minimum of 2 attribute instances per leaf.

2.6.3 Random Tree Classifier

Similar to the J48 classifier, the Random Tree classifier generates a decision tree. Unlike the J48 classifier, Random Tree does not ‘prune’ or optimise the resulting tree. This produces a significantly larger tree size, however training takes much less time. Using a much larger tree may become restrictive when storing and processing the classifier on a small embedded computer system. Due to this reason, research has been conducted on determining the length of the longest path in a binary search tree [17]. It has been discovered that a binary tree with n nodes will have the following longest path:

$$\frac{1}{\beta} \log n \approx 4.311 \log n$$

Where β is a unique number in the range $0 < \beta < 1$ satisfying the equation:

$$2\beta e^{1-\beta} = 1$$

Nodes are inserted one at a time randomly, as opposed to the J48 classifier which assesses the normalised information gain to narrow down the tree size during training. When used in this research, WEKA is instructed to generate a Random Tree classifier with a minimum of 1 attribute per leaf and seed the random number generator with ‘1’.

2.6.4 Neural Network Classifier

The Neural Network classifier is a biologically-inspired method, in particular, by the brain. The system is presented as a network of ‘neurons’ in a chosen number of ‘layers’. Each neuron may have a particular weighting or bias from a particular input, outputting a certain value only when certain inputs are met. A simple single-layer neural network is depicted in Figure 2-9. Five inputs to the network are connected to one output through a single ‘hidden’ layer, consisting of three nodes.

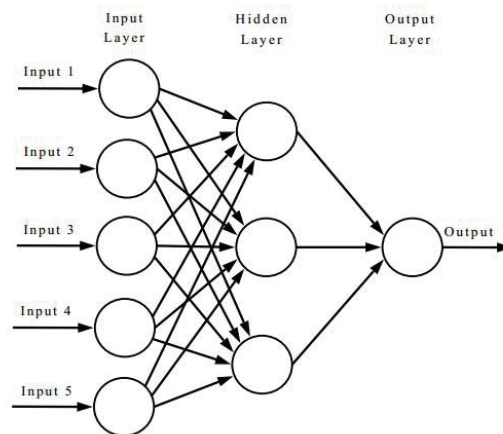


Figure 2-9: Single Layer Neural Network

Figure Created by Julien Cretel, StackExchange

The system trains by passing in an expected value and assessing the output. The most common method is by using a back-propagation algorithm to update the weights for each node in the network. This weight-updating rate, known as the learning rate, directly effects the training speed compared to quality.

The following process is used to train a simple neural network with back-propagation:

Algorithm 2-2: Neural Network Back Propagation

Phase 1: Propagation

Propagate the training signal through the network,
activating the output nodes.

Back propagate the training signal, generating delta ‘error’ values for
the hidden and output nodes.

Phase 2: Weight Update

For each weight;

Calculate the weight gradient by multiplying the output delta
and input activation.

Subtract the learning rate ratio of the gradient from the weight.

A network with a single hidden layer, has the limitation of only ‘learning’ a function which does not require abstract features. Training a neural network to identify a model of car based on colour, size and number of wheels, for example, will require multiple hidden layers.

It is expected that the neural network will be one of the top performers in this research, even as a simple single layer network. This is due to the binary classification nature - something a single layer network excels in. When used in this research, WEKA is instructed to train the Neural Network at a learning rate of 0.3 with a back propagation momentum rate of 0.2 for 20 epochs.

2.7 Summary

A brief overview has been given to frameworks used in this paper - colour and the theoretical layout in the RGB colour space, the property of image exposure, perspective transformations and Fuzzy Logic. Four classifiers are to be used to judge the effectiveness of the MIDECC system. These classifiers have been chosen for being internationally recognised as standard machine learning methods, covering a range of complexities.

Chapter 3

3. Review of Related Literature

The human eye identifies colours quickly based on surrounding perceptions and lighting. This task is difficult for a computer vision based solution, as one colour may appear different hues under disparate conditions. Previous research has been conducted in an effort to both identify lighting conditions and the potential adjustments to clarify colours. In this chapter, research that looks at the main components of classification, adjustment and transformation are reviewed.

3.1 Colour Classification

Recognising colours requires a reliable classification process which includes the ability to account for similar colours and varying illumination [18]. Additionally, a proposed system must allow for small intricacies that occur in differing cameras such as the behaviour to reflected light, noise, lens glare [19] and even quantum electrical effects in the camera sensor chip [1]. The following research examines different colour classifiers for use in varying conditions.

3.1.1 Robustness of Colour Detection for Robot Soccer

P J Thomas, Russel J Stonier and P J Wolfs, 2002 [19]

The ‘pie slice’ classifier is introduced in this paper as an alternative colour classification method. The use of standard rectangular colour ‘decision regions’ (Figure 3-1) may result in misclassifications if the luminosity varies slightly from the trained values. It is stated that a preferred classifier would be ‘insensitive to colour drift and glare’.

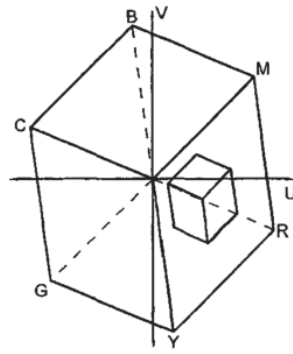


Figure 3-1: Rectangular decision region in the YUV colour space

In the YUV colour space, changes to the colour temperature would result in the colour rotating around the Y-axis, while changes to the saturation would adjust the distance of the point from the Y-axis.

Thomas et al., presented the pie slice classifier an alternative classifier that ignores the luminance component (Y) in this colour space.

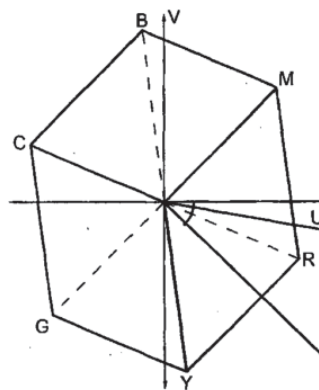


Figure 3-2: Pie Slice classifier in the YUV colour space

In Figure 3-2, an angle and radius value constructs a 'pie slice' inside the selected colour space. The angle ranges define a set of hues, while a minimum radius value ensure saturation is limited.

Thomas et al., found that if the radius is set too low, too many objects are classified in the playing field. Setting the radius too high resulted in the opposite - the number of classified pixels would decrease, insufficient to perform identification.

Interestingly, in the experimental observations it is noted that the pie slice classifier has specific strengths. Identifying blue, yellow and pink colours, even in changing lighting conditions from warm-temperature quartz-halogen to cooler fluorescent lighting caused no issues. It was also mentioned that the pie slice classifier is able to identify multiple shades of green successfully, a feature that will be explored further in this research.

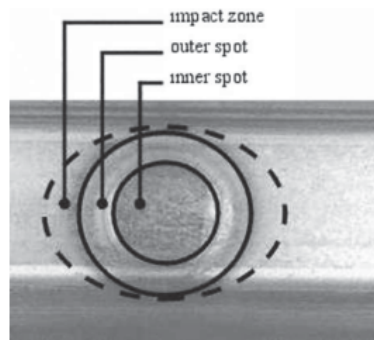
If $((\Phi_{uv} > \Phi_1) \&\& (\Phi_{uv} \leq \Phi_2) \&\& (dist_{uv} > r))$ Then (colour = 1)

The above formula is an example of a simple classifier proposed by Thomas et al., classifying a colour if an input angle and radius match defined ranges. In this formula, Φ represents the input angle, while $dist$ represents the distance from the centre of the colour space. This simplistic style of classifier will be used extensively in this research, as it enables simply binary-like matching of colours.

3.1.2 Knowledge-Based Fuzzy Colour Processing

Lars Hildebrand and Madjid Fathi, 2004 [20]

This paper explores a knowledge-based approach, generating a fuzzy logic based system to assess the quality of resistance spot welding. This proposed system identifies the size, colour and shape of three different regions of interest (ROI) in an input image.



QUALITY CRITERIA FOR GOOD AND POOR WELDING SPOTS

Quality Criteria ^a	Good Quality	Poor Quality
size of inner spot	8 mm	7.5 mm
size of outer spot	13.8 mm	9.9 mm
size of impact zone	18.7 mm	10.7 mm
shape of inner spot	circular	elliptical
shape of outer spot	circular	elliptical
shape of impact zone	circular	circular
color of inner spot	blue - dark blue	light grey - light red
color of outer spot	light blue	blue - red
color of impact zone	red - dark red	red - light red
size of inner spot	8 mm	7.5 mm

^aThe size of the spots depend on the thickness of the sheets as well as on the diameter of the electrodes. The thickness in this example is 3 mm, and the diameter of the electrodes is 8 mm.

Figure 3-3: ROI for a Resistance Spot Weld (left),
Table depicting Quality Criteria for comparing good and poor welding spots (right)

The above figure and table outlines the regions of interest, with the quality criteria to be examined. The paper explains the use of the HSI colour model, highlighting advantages such as the grouping of colours similar to the human eye, with a clear distinction between grey and saturated colours. Converting the three components of red, green and blue to the HSI model utilises the following definitions:

$$I_u = \left\| \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\| = \begin{pmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{pmatrix} \quad R = \begin{pmatrix} 2 \\ -1 \\ -1 \end{pmatrix} \quad R_u = \|R\| = \begin{pmatrix} \frac{\sqrt{6}}{3} \\ -\frac{\sqrt{6}}{6} \\ -\frac{\sqrt{6}}{6} \end{pmatrix} \quad G_u = \|I_u \times R\| = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

Where I_u , R , R_u and G_u are needed to define unit vectors for transformation.

Next, an extended version of the $atant(x,y)$ function is introduced:

$$atant(x,y) = \begin{cases} atanx(x,y), & \text{if } atanx(x,y) \geq 0 \\ atanx(x,y) + 2\pi, & \text{else} \end{cases}$$

Where

$$atanx(x,y) = \begin{cases} atan\left(\frac{x}{y}\right), & \text{if } x > 0 \\ atan\left(\frac{x}{y}\right) + \pi, & \text{if } x < 0, y > 0 \\ atan\left(\frac{x}{y}\right) - \pi, & \text{if } x < 0, y < 0 \end{cases}$$

And

$$atanx(0,y) = \begin{cases} 0, & \text{if } y = 0 \\ \frac{\pi}{2}, & \text{if } y > 0 \\ -\frac{\pi}{2}, & \text{else} \end{cases}$$

After these values have been defined, the transformation to HSI model from RGB values may now be expressed as:

$$C_{HSI} = \begin{pmatrix} H(C_{RGB}) \\ S(C_{RGB}) \\ I(C_{RGB}) \end{pmatrix}$$

With

$$C_{RGB} = \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

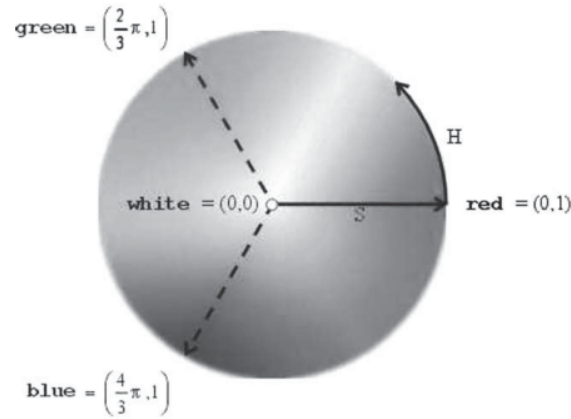
And

$$H(C_{RGB}) = atant(C_{RGB} \cdot G_u, C_{RGB} \cdot R_u)$$

$$S(C_{RGB}) = 1 - \frac{\min(C_{RGB})}{I(C_{RGB})}$$

$$I(C_{RGB}) = \frac{1}{\sqrt{3}} C_{RGB} \cdot I_u$$

These calculations result in Hue, Saturation and Intensity values which describe the same colour as C_{RGB} .

Figure 3-4: HSI Model for $I = 1.0$

Removing a dimension from the 3D model enables Hildebrand and Fathi to introduce the Fuzzy membership functions, highlighted in Figure 3-5. Removing the intensity value is accomplished by setting this value to a constant 1.0, while removing the saturation value sets this value to a constant 1.0.

Removing the intensity coordinates when lighter colours are important or the saturation coordinates when darker colours are important creates unique HS and HI-colour spaces. The HS-colour space is a subspace of HSI where the Intensity value is fixed at 1.0, while the HI-colour space is a subset where the saturation value is fixed at 1.0.

Incorporating a fuzzy set over the HS-colour space is defined by eight points. Each corner of the fuzzy set is represented by a point, depicted below.

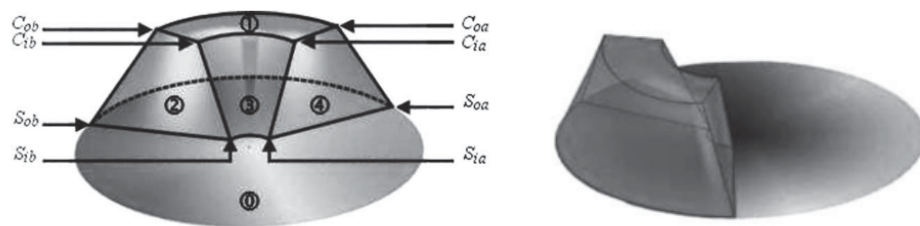


Figure 3-5: Fuzzy set, defined over the HS-colour space (left), Definition of the colour red (right)

The points are named using the terminology ‘support’ and ‘core’, with the letters I and O for inner and outer, respectively. Hildebrand and Fathi successfully create a process which incorporates a fuzzy logic system and unique colour spaces. The performance of the proposed system excels at correctly assessing quality welds, the first system developed to run without human supervision.

The below data plots show the frequency distributions of good samples compared to poor samples.

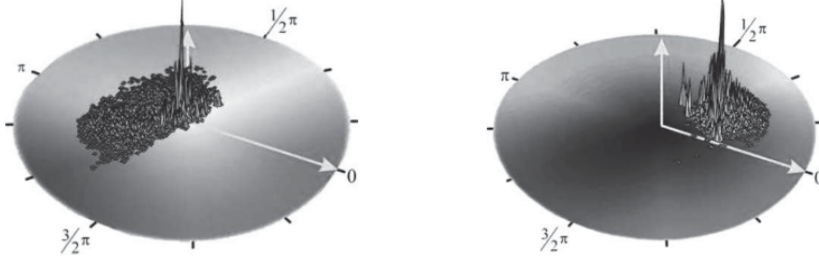


Figure 3-6: Frequency distribution of a good sample (left) compared to a poor sample (right)

3.1.3 Colour Classification using margin-setting with ellipsoids

Kaveh Heidary and H. John Caulfield [21]

The genetic algorithm presented in this paper designs ellipsoids which group pixels by a certain colour, partitioning the image. Once the image has been partitioned, classifiers are generated which are able to identify similar colours quickly with high accuracy.

The process, introduced as margin-setting, identify two parts of each colour. The two training parts are split by areas which can be classified by a large margin - easier to identify, while the other area is comprised of points which do not classify well. Areas which do not train well may include edges of different coloured objects, or similar coloured objects placed next to each other.

This paper uses ellipses in the RGB colour space, comprising of a foci-pair (\vec{F}_1, \vec{F}_2) and a radius parameter R . An arbitrary vector is classed as ‘inside’ a particular ellipsoid if the sum of its distances to the foci-pair is less than the ellipsoid radius R .

The equation below describes a vector-ellipsoid relationship, used to calculate the relationship between different classes during classification.

$$D_{\vec{V},e} = \sum_{k=1}^2 ||\vec{V} - \vec{F}_k||$$

Where $D_{\vec{V},e}$ represents the vector-ellipsoid distance, \vec{V} is the arbitrary RGB vector, $\vec{F}_k (k = 1,2)$ are the foci-pair of the ellipsoid and $||\vec{X}||$ denotes the L2-norm or Euclidean distance of \vec{X} .



Figure 3-7: Result of filtering the source image into class one (left) and class two (right) with a zero-margin classifier

Above, the result of filtering a simple scene with two distinct ‘classes’ is shown. In this example, one class - the leaf, has been correctly separated from the other class - the background books. Near the edge of the leaf in the right-hand ‘class two’ image, small areas of green can be seen where the algorithm struggled. Also note the light green rectangles in the left-hand ‘class one’ image - a similar colour to accents in the leaf, however these areas have been identified in the book background, incorrectly.

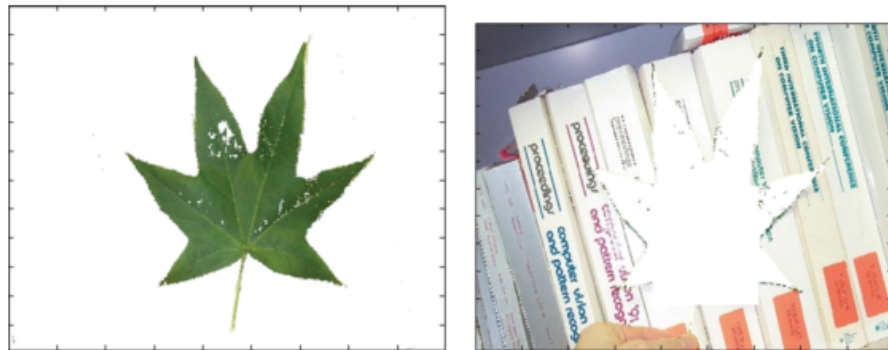


Figure 3-8: Result of filtering the source image into class one (left) and class two (right) with a 25% margin classifier

The above comparison with a 25 percent margin classifier results in less leaf-edge from showing on the right-side ‘class two’ image, with no background light green ‘bleed’ from the books in the class one image. Note there are some pronounced gaps in the middle of the leaf, where the colour classification system has misclassified the classes. Heidary and Caulfield state this indicates the systems’ ‘indecision’, where the higher margin resulted in pixels not captured by the filter.

In conclusion, this paper presenting margin-setting with ellipsoidal decision surfaces is a robust and computationally efficient, even when providing a minimal number of training samples for each class.

The setting of a margin value can vary the required ‘certainty’ of each pixel, however setting this value too high results in more unclassified pixels. It is suggested that groups of unclassified pixels or ‘holes’ in the image may be filled with techniques such as median filtering or mathematical morphology.

3.2 Contrast Adjustment

Adjusting the contrast of the input image enables a proposed system to ‘push’ or ‘pull’ colours into a classifier [1]. If implemented correctly, adjustment of contrast may be set to enable a wider range of colours or counter texture or illumination differences. The following research papers explore adjustment of contrast for the use of expanding classification results.

3.2.1 Colour image enhancement using correlated intensity and saturation adjustments

Ngaiming Kwok, Haiyan Shi, Gu Fang, Quang Ha, Ying-Hao Yu, Tonghai Wu, Huaizhong Li and Thai Nguyen (2015) [22]

Two approaches to contrast adjustment are presented in this paper to emphasise object edges and saturation. The process has been tuned to only boost saturation in regions of interest.

The first proposed algorithm, named ‘Intensity and saturation adjustment by intensity and saturation’ (ISAIS) employs two separate formulae to calculate the output intensity and saturation:

$$\begin{aligned} I_{enh}(u, v) &= I_{in}(u, v) + k_1 \Delta I(u, v), \\ S_{enh}(u, v) &= S_{in}(u, v) + k_2 |\Delta S(u, v)|, \end{aligned}$$

where k_1 , k_2 are gain factors for the intensity and saturation, respectively. The differences between input intensity and saturation attributes and local average are calculated as below:

$$\begin{aligned} \Delta I(u, v) &= I_{in}(u, v) - \bar{I}_{in}(u, v), \\ \Delta S(u, v) &= S_{in}(u, v) - \bar{S}_{in}(u, v), \end{aligned}$$

where the local averages are derived from averaging pixel magnitudes in a square patch of size 3x3 pixels.

The second approach, a refinement to the above algorithm titled ‘Intensity and saturation adjustment by intensity and saturation correlation’ (ISAISC) uses the correlation between the intensity and saturation values, resulting in more effective output values.

$$I_{enh}(u, v) = I_{in}(u, v) + k_1 \Delta I(u, v) + k_2 \rho(u, v) \Delta S(u, v),$$

$$S_{enh}(u, v) = S_{in}(u, v) + k_3 \Delta I(u, v) + k_4 |\rho(u, v) \Delta S(u, v)|,$$

where the correlation value $\rho(u, v)$ is calculated with the below formula,

$$\rho(u, v) = \frac{\sum_{\Omega} \Delta I(u, v) \Delta S(u, v)}{\sqrt{\sum_{\Omega} \Delta I^2(u, v) \sum_{\Omega} \Delta S^2(u, v)}}$$

where Ω is reported to be 3, creating an average area of 3x3 pixels for further examples.

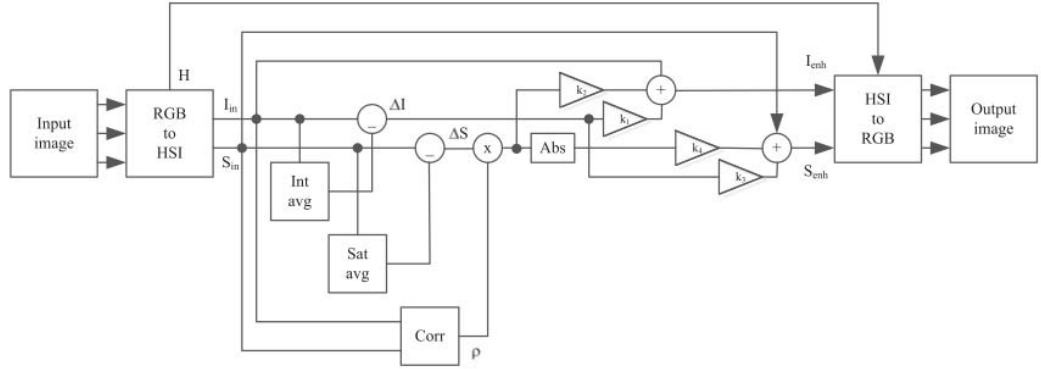


Figure 3-9: ISAISC General System Architecture

Seen in the above ISAISC block diagram, four scale factors - k_1 to k_4 control strengths of adjustment. For the below examples, all scale factors are set to a value of 2.0.



Figure 3-10: Source Image (top), ISAIS Result (left), ISAISC Result (right)

The outputs from the proposed approaches highlight the rock texture closer to the camera, while also highlighting the far end of the house. The ISAISC results in more detail being highlighted in the garden to the far left, however it is noted that ISAISC is computationally more expensive due to the correlation calculations. It is stated by Kwok et al., that the difference between the two methods largely rests on the time-consuming correlation calculations - ISAIS provides an acceptable middle ground between speed and quality, while ISAISC provides the highest quality processing.

3.2.2 Dynamic Colour Object Recognition Using Fuzzy Logic

Napoleon H. Reyes and Elmer P. Dadios (2003) [1]

This research paper proposes a fuzzy logic-based algorithm to enhance or degrade the red, green and blue channels of an input image. Additionally, the paper introduces the previously discussed pie slice classifier and its' use with the normalised rg-chromaticity space.

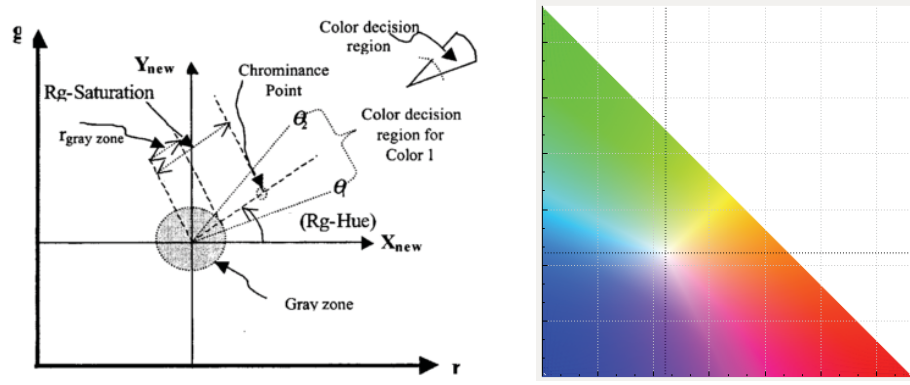


Figure 3-11: Pie Slice colour decision region in rg-colour space (left), rg-colour space (right)

The normalised rg-chromaticity colour space, above right, features heavily in this paper. The colour space layout reduces the effect of brightness variation, however when combined with the pie slice classifier, negative effects such as hue-drift and misclassifications due to changes in brightness are considerably reduced.

Deriving the pie-slice angle and radius in relation to the normalised rg-chromaticity (r, g) colour space from RGB values are as follows:

1. Compute the normalised rg-chromaticities from the input RGB values

$$r = \frac{R}{R+G+B} \text{ and } g = \frac{G}{R+G+B}$$

2. Assigning white, located at (0.333, 0.333) as the origin, compute the rg-Saturation (radius) and rg-Hue (angle) as a point extending for any given chromaticity.

$$rg \cdot Saturation = \sqrt{(r - 0.333)^2 + (g - 0.333)^2}$$

$$rg \cdot Hue = \tan^{-1}((g - 0.333)/(r - 0.333))$$

Discussed in previous research, Reyes and Dadios highlight the pie-slice classifiers' ability to envelope neighbouring colours and reduce the effects of glare.

This is further complemented by using the normalised rg-chromaticity colour space, however the importance of excluding a 'grey zone' is stressed.

As both the colour space and pie-slice classifier reduce the effects of brightness, plotting high or low brightness data points into the colour space may produce unexpected results.

Once the system is trained, Reyes and Dadios use a look-up table to convert each angle and radius pair to a corresponding colour. The lookup table is generated for all possible colours using rules of the form:

If ($rg.Saturation \geq r$) and ($rg.Hue \geq \theta_1$) and ($rg.Hue \leq \theta_2$) **Then** $Colour = 1$.

The paper describes a system which adjusts the red, green and blue values in order to adjust the colour hue and saturation. These adjustments are designed to enable the system to identify colours in a larger range of illumination. The two complementary colour contrast adjustment operations are as follows:

1. Contrast Intensification Operator

This operator uses the Logistic function to alter the colour channel contrast. This function intensifies an input signal closer to 1.0, increasing the contrast.

$$\alpha = \begin{cases} 2\mu_{\alpha}^2(y) & 0 \leq \mu_{\alpha}(y) < 0.5 \\ 1 - 2[1 - \mu_{\alpha}(y)]^2 & 0.5 < \mu_{\alpha}(y) \leq 1 \end{cases}$$

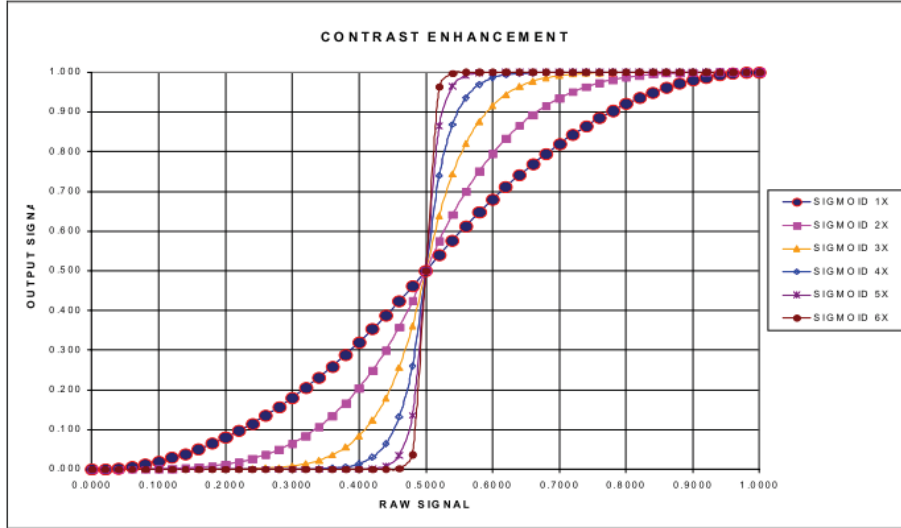


Figure 3-12: Contrast intensification operator

Successive iterations of the logistic function creates a binary-like contrast adjustment. It is noted that the certain strengths of the enhancement may be suited to certain applications - scenes with greater dynamic range may require a higher level of adjustment, for example.

2. Contrast Degradation Operator

This operator uses the Logit function to alter the colour channel contrast. This function degrades an input signal towards half the threshold of 0.5, decreasing contrast.

$$\alpha = \begin{cases} 0.5 + 2[\mu_{\alpha}^2(y) - 0.5]^2 & 0 \leq \mu_{\alpha}(y) < 0.5 \\ 1 - (2[1 - [\mu_{\alpha}(y) + 0.5]]^2 - 0.5) & 0.5 < \mu_{\alpha}(y) \leq 1 \end{cases}$$

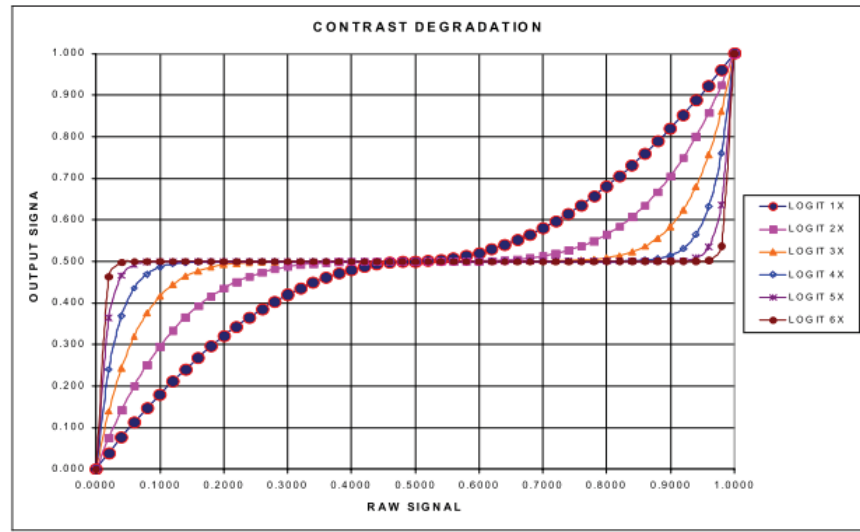


Figure 3-13: Contrast degradation operator

Complementary to the logistic function, successive applications of the logit function produce a stronger pull to the half-threshold value, seen above. The Logic-Logistic Fuzzy Colour Constancy algorithm presented in this paper uses fuzzy logic to apply different strengths of intensification or degradation to the input camera signal. This adjusted signal is then classified using the pie-slice classifier on the rg-chromaticity colour space, identifying a single colour.

The paper also mentioned that for the optimum mapping of membership functions, adjustments should be made precise up to at least one decimal place.

In conclusion, Reyes and Dadios have successfully built a system which uses a unique contrast adjusting algorithm. This algorithm successfully identifies light blue, blue and violet at 96%, 94% and 72% respectively. It is reported that the techniques presented show robust classification under spatially varying illumination, object position and rotation.

3.3 Image Exposure Selection

There are several methods of choosing an ideal camera exposure, some of which are researched below. Papers were chosen based on their automatic exposure finding process - research which includes unknowns such as the lack of external reference points are included, as this represents a goal of this research. It is expected that this research proposal will allow for automatic exposure calibration with minimal reference points or previous knowledge.

3.3.1 Acquisition of Agronomic Images with Sufficient Quality by Automatic Exposure Time Control and Histogram Matching

Martín Montavlo, José M. Gurrero, Juan Romeo, María Guijarro, Jesús M. de la Cruz and Gonzalo Pajares (Advanced Concepts for Intelligent Vision Systems, 15th International Conference AACIVS 2013, Poznań, Poland, October 2013) [2]

This research focuses on adjusting the exposure time of a Kodak KAI 04050M/C sensor in order to accurately detect crop lines and identify weeds in an agricultural setting. Proposed is a system which analyses an image histogram to provide the exposure adjustment. The system also accounts when higher-exposure images are taken from a moving tractor, resulting in a blurry, unstable input.

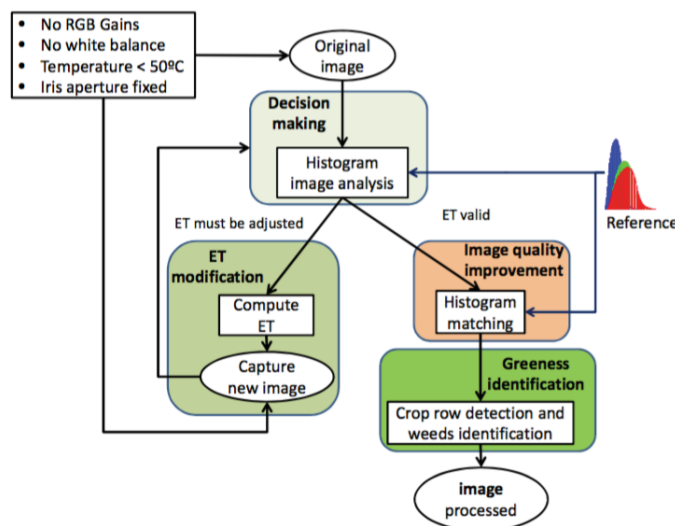


Figure 3-14: General Architecture of the Automatic ET Control and Histogram Matching System

The block diagram of the proposed system highlights a fixed reference histogram which is introduced to the ‘decision making’ block where the histogram analysis is run.

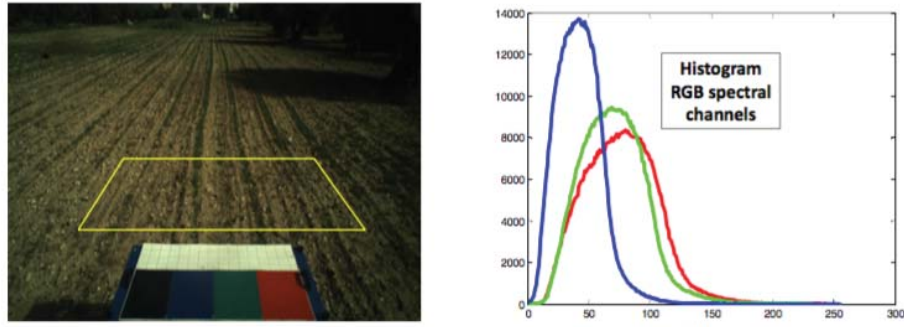


Figure 3-15: Input Image highlighting the ROI (left), Histogram for the RGB channels in the ROI (right)

The use of the histogram has been found to provide an accurate representation of image quality. A histogram with concentration in the low or high regions indicates under exposure or over exposure, respectively. Letting g be a random variable denoting grey levels, where the n th moment of g about the mean is defined as:

$$\mu_n(g) = \sum_g (g - m)^n p(g) \text{ with } m = \sum_g g \cdot p(g)$$

This equation assists in further analysis of the ‘most important statistical parameters’ to assess image quality; mean [m], variance [$v = \mu_2(g)$], skewness [$\gamma = \mu_3(g)\mu_2^{-3/2}$] and kurtosis [$\kappa = \mu_4(g)\mu_2^{-2}$]. While the commonly used mean and variance are used to assess the average and distribution of values, the skewness measures the asymmetry of the distribution, while the kurtosis value indicates the ‘peakedness’ of values. In this paper, the mean and skewness are the only values found to be needed for the successful Exposure Time decision making process. These values are used as follows;

While $(m_i \geq m_{ui} \text{ and } \gamma_i \geq |\gamma_{ui}|)$ or $(m_i \leq m_{ii} \text{ and } \gamma_i \geq \gamma_{ii})$

$$ET = \left(\frac{m_{Rref} + m_{Gref}}{m_R + m_G} \right) ET_{curr}$$

Otherwise $ET = ET_{curr}$

The empirically selected Exposure Time (ET) value is run twice, once for the red and once for the green channel. The specific agricultural application researched in this paper does not require the use of the blue channel in the region of interest.

A total of 1,358 images were processed in this paper, while the tractor moved at approximately 4 km/h. The images were taken with varying exposure times for adjusting parameters.



Figure 3-16: Images captured with different exposure times for adjusting parameters

Each of these images are analysed in their respective ROI until the highest possible degree of satisfaction is established, scored by the above Exposure Time decision making process. Once this has been completed, the reference histogram is saved for future adjustments. This paper then successfully continues to identify crop lines and weed patches in the region of interest, however an emphasis is placed on the method of exposure selection for this research.

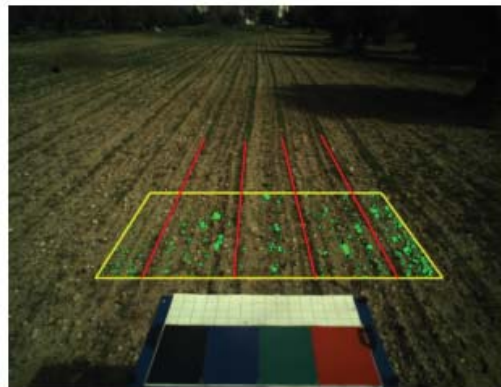


Figure 3-17: Identification of crop lines and weed patches in the ROI

3.3.2 OCR Accuracy Improvement on Document Images through a novel pre-processing approach

A. El Harraj and N. Raissouni (2015) [3]

This paper focuses on adjusting illumination for optical character recognition. The research experiments within the RGB, LAB, HSV and YUV colour spaces to estimate brightness levels and calculate possible brightness and contrast adjustments. The research also sharpens the image then uses Otsu thresholding to remove the background.

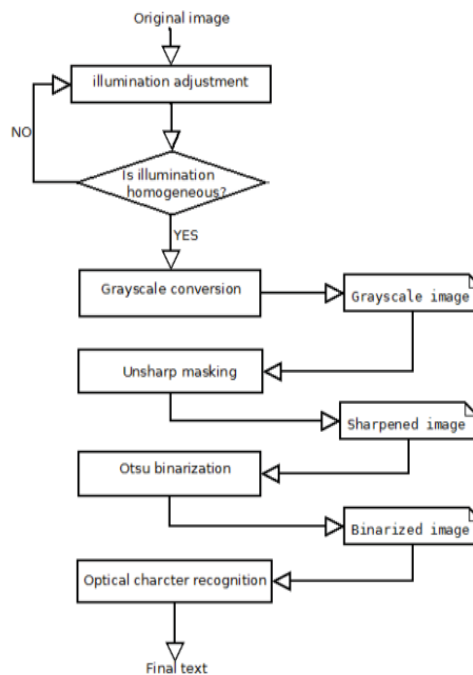


Figure 3-18: Block Diagram of the OCR pre-processing approach

The above system diagram depicts the individual steps, including the recursive illumination adjustment. It is stressed that the proposed solution is suitable for any kind of document.

El Harraj and Raissouni have previously discussed a process called ‘Contract Limited Adaptive Histogram Equalisation’ (CLAHE) in an earlier paper as a method of making hidden features of an image more visible. CLAHE was originally developed for medical imaging, specialised from Adaptive Histogram Equalisation [23] where background image noise may be incorrectly amplified.

Discussed is the use of Value in the HSV colour space in order to gauge an image brightness. It is stated that this is inaccurate for brightness estimation due to the value of 1.0 for white, as well as any 'pure colour'. The research states that for the first stage of illumination adjustment, CLAHE is used to enhance the areas of interest. In this research, the areas of interest include the edges of text on a page.

This process results in the below right-hand image, compared to the input image on the left.

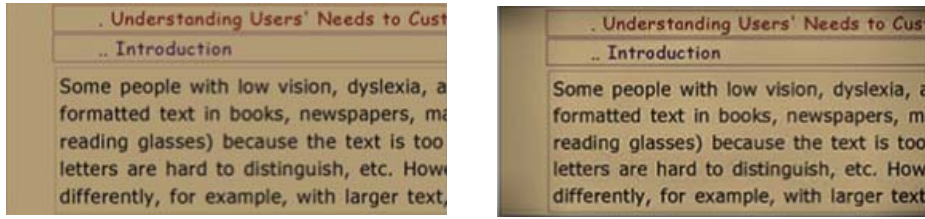


Figure 3-19: Original Image (left), Enhanced image using CLAHE equalisation (right)

The next stage of the system processing is to estimate the brightness of the image. El Harraj and Raissouni found Luma (Y') from the $Y'UV$ colour space provides the best estimate of brightness, using the weighted average of gamma-corrected R, G and B values. The relationship between $Y'UV$ and RGB can be seen in the below matrix:

$$\begin{bmatrix} Y' \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ -0.14713 & -0.28886 & 0.436 \\ 0.615 & -0.51499 & -0.10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Which may be simplified to:

$$Y' = 0.299 * R + 0.587 * G + 0.144 * B$$

The last pre-processing step implements the Otsu Binarisation approach. A limitation of the Otsu algorithm is the assumption of uniform illumination. By using the Luma (Y') value calculated above, the paper proposes a simple and efficient pixel transform to create an operator for brightness and contrast adjustment. This transform is represented by the equation:

$$g(x, y) = \alpha \cdot f(x, y) + \beta$$

Where $f(x, y)$ is the source image, $g(x, y)$ is the processed image.

Using the Luma value calculated previously, the paper states the proposed approach uses $\alpha = 1.4$ and $\beta = 50$.

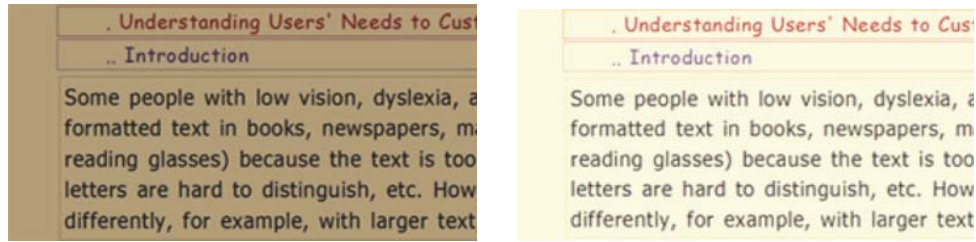


Figure 3-20: Original Image (left), Enhanced image using the proposed brightness and contrast adjustments (right)

For text recognition, images involving intensity values only are adequate for identification. Grey scale representations are often used for extracting descriptors as it simplifies the algorithm and reduces computational requirements. This paper uses the luminance algorithm, shown below:

$$\text{Luminance} = 0.3 * \text{Red} + 0.59 * \text{Green} + 0.11 * \text{Blue}$$

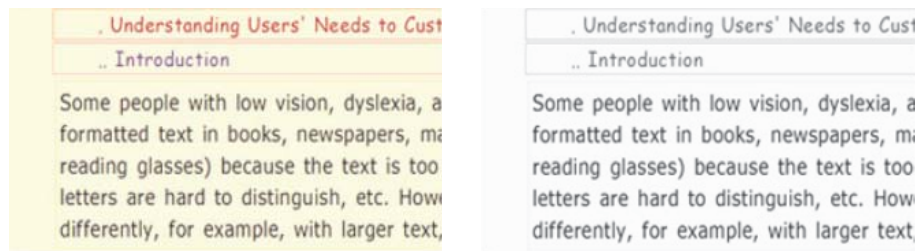


Figure 3-21: Image after contrast and brightness adjustment (left), Result of luminance algorithm for grey scale conversion (right)

With the pre-processing complete, the next step uses an Un-sharp masking filter to enhance text details. Also known as an edge enhancement filter, the un-sharp masking filter is a simple operator which subtracts a blurred image from the original.

Blurring the image using a Gaussian filter with a reported size of 3x3, the image is then run through the un-sharp masking filter which is built into the OpenCV library.

The reported values of Amount = 1.5, Radius = 0.5 and Threshold = 0, produce the following output:

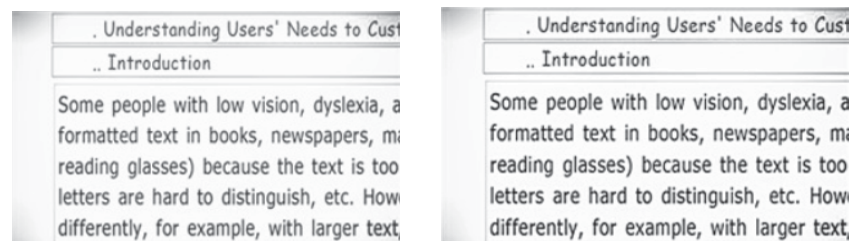


Figure 3-22: Grey scale image (left), Result of un-sharp masking filter (right)

Un-sharp masking is very successful at sharpening images, however it is noted that with incorrect values, these results may be detrimental producing undesirable effects including artefacts near edges.

The last step of image processing before being sent to the Optical Character Recognition system is cleaning or removing the document background. A simple threshold requires fine tuning in order to successfully classify each pixel as an object or background. This fine tuning would result in variances across a high dynamic range picture, resulting in misclassifications. For this reason, El Harraj and Raissouni implemented Otsu's thresholding method to minimise the interclass variance of thresholded pixels [24].

$$\sigma_w^2(t) = q_{1(t)} \sigma_1^2(t) + q_{2(t)} \sigma_2^2(t)$$

where

$$\begin{aligned} q_1(t) &= \sum_{i=1}^t P(i) & q_2(t) &= \sum_{i=t+1}^I P(i) \\ \mu_1(t) &= \sum_{i=1}^t \frac{iP(i)}{q_1(t)} & \mu_2(t) &= \sum_{i=t+1}^I \frac{iP(i)}{q_2(t)} \\ \sigma_1^2(t) &= \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} & \sigma_2^2(t) &= \sum_{i=t+1}^I [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)} \end{aligned}$$

The result of running the Otsu algorithm on the previous steps' sharpened output is below:

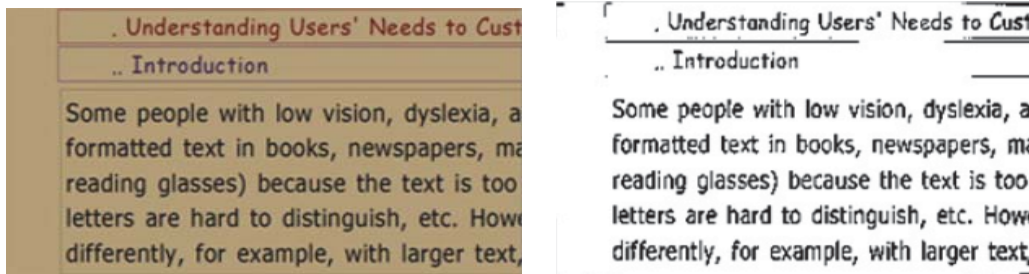


Figure 3-23: Grey scale sharpened image (left),
Result of binarised image using Otsu thresholding approach (right)

Once the above processing has been completed on each image, it was then sent to *Tesseract-OCR* software to compare against the original image. The paper reports a 2% - 6.8% increase in text classification accuracy.

3.4 Multiple Camera Input and Processing

Receiving multiple input images enables a computer vision system to see multiple angles or exposures. A processing system may then ‘choose’ which image is preferred depending on certain criteria. The following research explores some examples of multiple camera input and different processing methods. For brevity, just the input architecture of the proposed systems is studied.

3.4.1 A unified framework for multi-sensor HDR video reconstruction

Joel Kronander, Stefan Gustavson, Gerhard Bonnet, Anders Ynnerman and Jonas Unger (2014) [6]

Presenting a novel High Dynamic Range (HDR) image sensor, this paper does not require alignment of input images as the single input lens houses three individually tuned and processed Kodak KAI-04050 sensors. Each sensor receives a fraction of the input light, depending on the beam splitter layout and neutral density (ND) filter. A filtering framework is proposed which addresses all of the commonly found issues with HDR pipeline processing simultaneously in a formalised way.



Figure 3-24: Common HDR System processing steps

Kronander et al., states the vast majority of previous HDR treat the following processes as separate tasks, with input from each Low Dynamic Range (LDR) sensor:

- *Demosaicing* of a colour filter array (CFA) sampled data, reconstructing the full colour image from the LDR sensor data
- *Resampling* which corrects for any geometric misalignments between each LDR sensor
- *HDR Assembly*, an important step where regions of each LDR image are selected for output in the final image
- *Denoising* which reduces image and colour noise

The paper notes that there are several issues with the above pipeline architecture:

- Performing demosaicing before the HDR Assembly step increases bad or missing data around saturated pixels
- Performing demosaicing after the HDR Assembly increases blur and ghosting unless the sensors are perfectly aligned.

Aligning multiple high-resolution sensors is reported to be problematic and costly, especially if attempting to align each with the precision required for the CFA patterns to match for the first processing step of Demosaicing.

While this paper continues to discuss the use of calibration and combining the LDR images to a HDR output, the points regarding the multiple camera pipeline architecture are the main contributions to this research.

3.4.2 Object Tracking in the Presence of Occlusions Using Multiple Cameras: A Sensor Network Approach

Ali O. Ercan, Abbas El Gamal and Leonidas J. Guibas (2013) [4]

This article proposes a unique method of reducing data transmission required by each camera, speeding up computation time. It is noted that current research attempts to compress the signal or perform sophisticated vision processing at each camera. While signal compression still requires a high speed of communication, processing at the camera requires a significant computational energy at the remote camera location.

Proposed is a task driven approach, where ‘simple local processing’ is performed at each camera, extracting only the essential information required. This substantially decreases the data to be transmitted from the camera while reducing the computational energy required.

This paper focuses on the efficient selection of the target object, removing other elements in order to reduce network traffic.

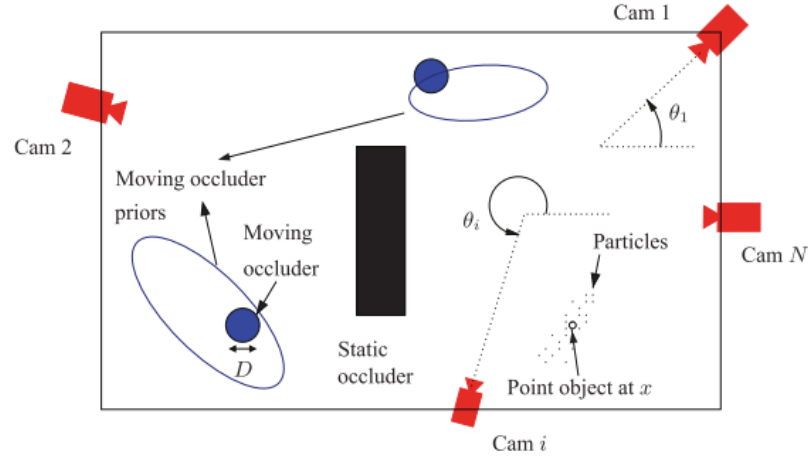


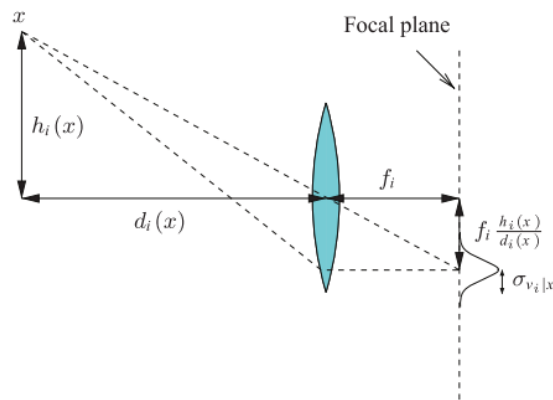
Figure 3-25: Illustration of the Object Tracking in the Presence of Occlusions problem setup

For each camera in Figure 3-25, it is assumed that:

- The roll angle (rotation around the optical axis) is zero
- The pitch angle (vertical tilt) is close to zero
- The cameras are mounted slightly above the height of an average human

Given these assumptions, the calibration for each camera consists of the camera coordinates and yaw angle (rotation around the vertical axis). It is also assumed that the object to track is a ‘point object’ - meaning it is possible to detect by some specific point features. There are to be M other moving objects, each modelled to a cylindrical shape of diameter D . It is also assumed that the identification of either ‘object to be found’ or the ‘moving occluder’ has been completed, as this paper focuses on minimising network traffic, compared to the actual identification. The positions and shapes of static occluders in the room are to be known in advance.

For each camera, background subtraction is performed locally as part of the object detection process.

Figure 3-26: Perspective Camera measurement model, with unoccluded object at x .

If the object has been found in the camera field of view, its horizontal position in the image plane is estimated and sent to the cluster head. It is noted that this is a very small amount of data compared to transmitting the whole image. This horizontal position of the object in the image plane is given by:

$$z_i = f_i \frac{h_i(x)}{d_i(x)} + v_i,$$

Where x is the location of the object, f_i is the focal length for camera i , and $h_i(x)$ and $d_i(x)$ are distances defined in the above figure. The value v_i is a value that allows for readout noise and calibration inaccuracies.

This paper explores the use of a moderate amount of processing at the camera ‘head’, enabling more cameras to be networked together without significant data congestion. This paper continues on to discuss detailed calculations to be run on the cluster head, however these methods are not applicable to the research needs of this paper so have been omitted from the review.

3.4.3 3D Scene Analysis by Real-Time Stereovision

Giovanni Garibotto and Carlo Cibeil (2005) [5]

This paper proposes a system that detects moving 3D objects across a plane. Scanning across the plane at different heights results in a 3D map being created by the use of homography and multiple cameras. Object segmentation and tracking is performed after the system has used 3D perception to detect change in an actual object, compared to a ‘phantom’ shadow or highlight.

Figure 3-27 depicts the processing pipeline for two cameras - indicated as a ‘master’ and ‘slave’. Interestingly only the master camera includes the motion and edge detection before being sent to the homography prediction and rectification process.

Garibotto and Cibeil state many current systems are limited by partial occlusions - when many people ‘hide’ others from the camera image.

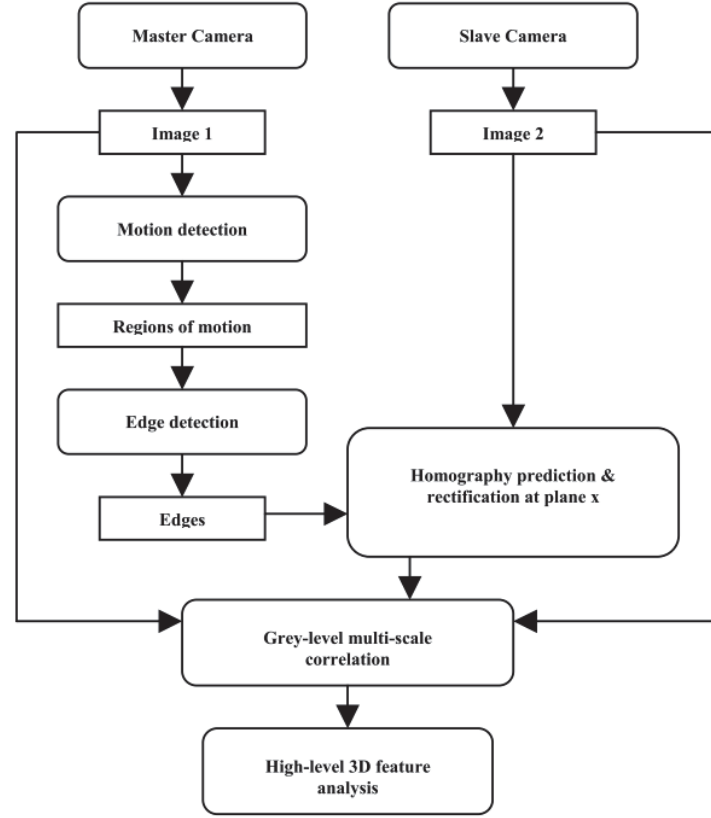


Figure 3-27: 3D Scene Analysis System General Architecture

The main objective of this research is to use a slave camera to create a ‘virtual projection’ of the scene.

Framing the scene by two cameras, the two images are related by the ‘epipolar constraint’, depicted below:

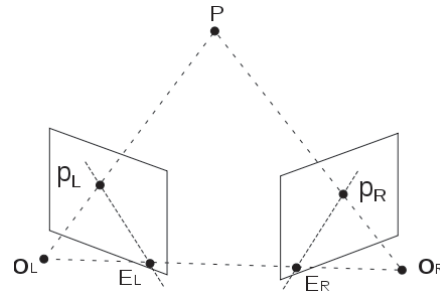


Figure 3-28: Example of Epipolar Geometry
 Figure created by Wikipedia user ‘ZooFari’

A point \mathbf{p}_1 in the master image maps to the point \mathbf{p}_2 in the second image, as:

$$k_2 \mathbf{p}_2 = k_1 * H_{\infty} * \mathbf{p}_1 + k_{2e} * \mathbf{e}_2$$

Where k_1 , k_2 and k_3 are real numbers, e_2 is the epipole of the master camera on the slave image. H_∞ is the homography matrix (at infinity), mapping the first image onto the second.

Homography is used to adjust the rotation between the two cameras, obtained using at least four reference points visible to each camera. Once the homographic matrix H_0 has been calculated using as many reference points as possible, the use of the infinite homographic plane H_∞ assists in obtaining all other homographic functions H_z at any other parallel plane as a linear combination:

$$H_z = \lambda_z H_0 + (1 - \lambda_z) H_\infty$$

Where $\lambda_z \rightarrow 0$ when $z \rightarrow \infty$ (plane at infinity) and $\lambda_z \rightarrow 1$ when $z \rightarrow h$ (reference plane). The best estimate of the relationship between the two cameras is given by $\lambda_z = h/z$, which leads to the interpolation equation:

$$H_z = h/z H_0 + (z - h) / z H_\infty$$

This interpolation equation may now be used to map the projected points along the epipolar line on the slave image. Using this approach, Garibotto and Cibeï state that the scene may now be ‘sliced’ by 3D parallel lines.

The remainder of the paper discusses calibration by using vanishing points that may assist in discovering the main calibration parameters. Once the vanishing point has been identified, it would be possible to recover the H_∞ matrix.

Vanishing points may be computed by identifying vertical and horizontal lines in the image. It is noted that this may not be the most precise identification, as it is very sensitive to small pixel noise.

3.5 Summary

Research has been reviewed which covers different methods of classifying colours, exposure selection, adjusting image contrast and the processing of multiple input images.

Classifying colour is a priority for this research, as a reliable classifier that has the capability to be somewhat resilient to changes in illumination.

The pie slice classifier introduced by P J Thomas, Russel J Stonier and P K Wolfs successfully ignores luminance in the YUV colour space. It is noted that this classifier has specific strengths, as identifying Blue, Yellow and Pink colours yield higher classification results compared to other colours. This classifier in use with the normalised r/g colour space has proven to provide a heightened resistance to changes in illumination. A basic 'binary' classifier is to be used to classify each colour, however the use of additional probabilistic calculations are to be explored for comparison between other standard machine learning algorithms.

Lars Hildebrand and Madjid Fathi introduced a unique Fuzzy Logic based solution to classify regions of interest in a metallic welding joint. The ability of the system to visually output the data points as a plot provides immediate feedback on quality, while the data points themselves may be used statistically, providing a greater level of feedback. The distribution of data points in a colour space is to be further examined in this research.

Kaveh Heidary and H. John Caulfield highlight the difficulties when detected edges of a brightly coloured object compare to the background. The algorithms' use of a 'certainty' value before a particular colour is classified, results in a system reduces ambiguity between two or more possible matches. This simplistic method of resolving conflicting disparate classifiers may be used later in this research to prioritise both colours and cameras.

Ngaiming Kwok and fellow researchers introduced two methods for adjusting the intensity and saturation of an image in specific regions of interest. The processes may be run on the whole input image, where adjustments will be made automatically to areas where details may be enhanced. It is noted that while providing a higher quality output, the ISAISC algorithm is computationally more expensive due to extra processing compared to the ISAIS. These methods have highlighted the importance of developing a computationally efficient algorithm. The strengths of the algorithms presented include the ability to enhance only the regions where additional noise will not be generated. The idea of an algorithm that intelligently 'skips' areas of non-interest will be explored later in this paper.

Discussed previously, Napoleon H. Reyes and Elmer P. Dadios paired the pie slice classifier with the red/green chromaticity colour space. The paper discusses a successful colour contrast adjustment process called 'Fuzzy Colour Contrast Fusion' (FCCF). It has proven a successful method of adjusting the red, green or blue channel of an input image in order to shift a colour into or out of a pie slice classifier. Overall, this process increases true positive matches, while reducing false negative classifications. While the process appears robust for a single input image, it is yet to be attempted with a multiple image input system. This research will attempt to implement the FCCF system in order to increase classification reliability.

Martín Montavlo and fellow researchers presented an automatic exposure adjustment solution powerful enough to run real-time from a slow moving vehicle with varying image qualities. The proposed system used histogram matching to heuristically adjust the exposure time of a single camera, requiring the use of reference points in the input image, or a previously known histogram reference. It is a goal of this research to automatically adjust the exposure with the target colours known only.

A. El Harraj and N. Raissouni presented automatic adjustment of the Value channel in the HSV colour space in order to clarify text on a page for an optical character recognition (OCR) program. The proposed process then uses the Y'UV colour space to estimate the brightness of the image, used in conjunction with a simple pixel transformation equation to adjust the brightness of the image. Techniques such as applying an unsharp-mask filter to increase clarity and Otsu thresholding to remove the background also assisted in the OCR program in increasing accuracy.

Joel Kronander and fellow researchers presented an overview of the current state of high dynamic range imagery processing, highlighting the various processing stages from a multiple low dynamic range images to a single high dynamic range output. While the paper does present a novel method of testing and introducing a new method of taking perfectly aligned LDR images from multiple sensors, these details were omitted from the review as it does not assist in the multiple camera colour-oriented system. This paper did emphasise the importance of processing steps - reordering the HDR assembly process produces substantially differing results.

Ali O. Ercan, Abbas El Gamal and Leonidas J. Guibas note the possibility of reducing data transmission across a camera network by running minimal processing on the camera before passing ROI coordinates to a 'cluster head' processor for further analysis. An emphasis is given on the known-values that are required for correct calibration. The lack of all known-values may result in a camera being unable to locate an object, producing erroneous results. While this paper focuses on reducing the traffic from the camera to the processing system, removing data early on from a colour classification system-processing pipeline may hinder results. The probabilistic approach to be used for the proposed colour classifier system requires the use of as much data as possible, so it is unlikely this method could be employed.

Giovanni Garibotto and Carlo Cibeï present calculations from which two cameras may work in tandem to create a 3D representation of a scene. Saving on computational power, only the dedicated 'master' camera includes processing steps such as motion and edge detection, while the 'slave' camera is used to calculate epipolar constraints. Homographic perspective processing will be used in the MIDECC system as a method of removing rotational differences between the three input cameras.

Many ideas have been explored, providing different methods to implement a colour classification system. The Pie Slice classifier will feature heavily in the proposed solution, while additional processing stages such as Fuzzy Colour Contrast Fusion and background removal will be explored further.

Chapter 4

4. Multiple Image Dynamic Exposure Colour Classification (MIDECC) System

This chapter introduces the Multiple Image Dynamic Exposure Colour Classification (MIDECC) System at the heart of this research. Each of the four main stages are discussed, from the image input and data sampling, to construction and final evaluation of the colour classifiers.

Unique systems are introduced, including an automated multi-camera exposure system, a complementary normalised cyan/magenta colour space and a simple yet effective method of resolving disputes between classifiers if multiple matches occur.

4.1 MIDECC General Architecture

The following diagram depicts the classification process used by the MIDECC System, highlighting the four unique processes novel to this research.

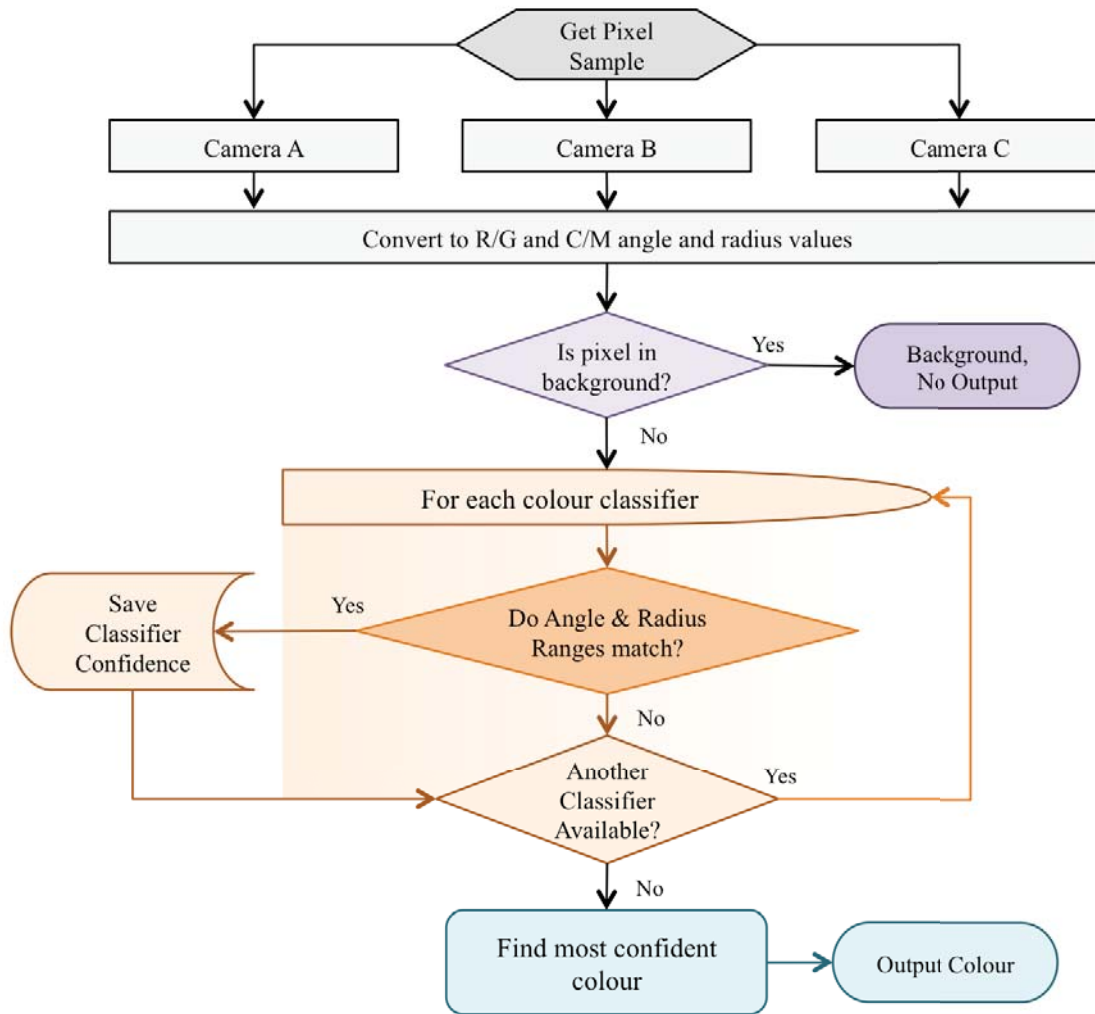


Figure 4-1: MIDECC System Classification Architecture

Camera Input	Background Detection	Colour Classification	Colour Space Selection
Section 4.3 Page 51	Section 4.6 Page 64	Section 4.5.3 Page 61	Section 4.5.4 Page 63

The following diagram depicts the training process used by the MIDECC system, splitting the process into four sub-processes that are explained in detail further in this chapter.

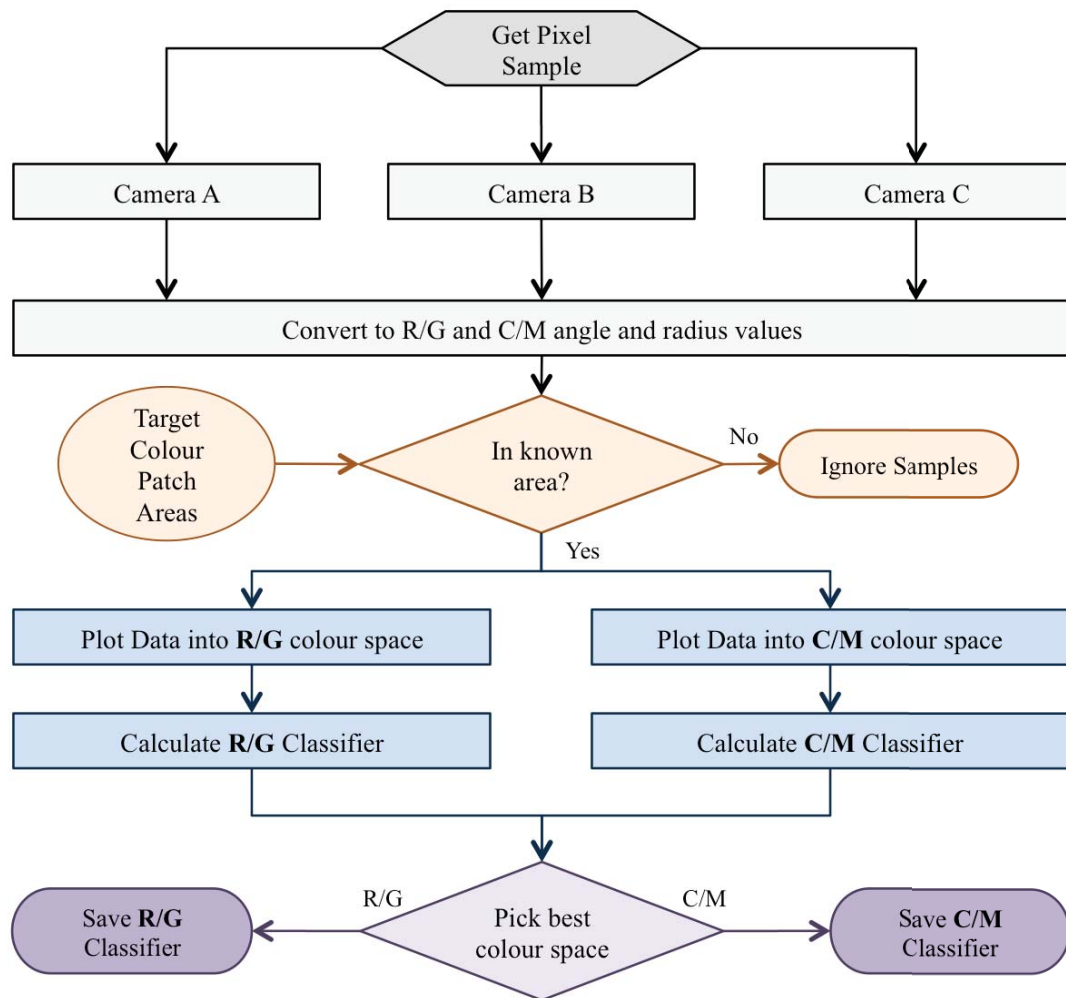


Figure 4-2: MIDECC System Training Architecture

Camera Input	Known-Area Training	Classifier Construction	Colour Space Selection
Section 4.3 Page 51	Section 4.4 Page 56	Section 4.5 Page 59	Section 4.5.2 Page 60

4.2 Introducing the Normalised Cyan/Magenta Colour Space

It has been noted in previous research the pie slice classifier has particular strength identifying blue, yellow and pink colours [19].

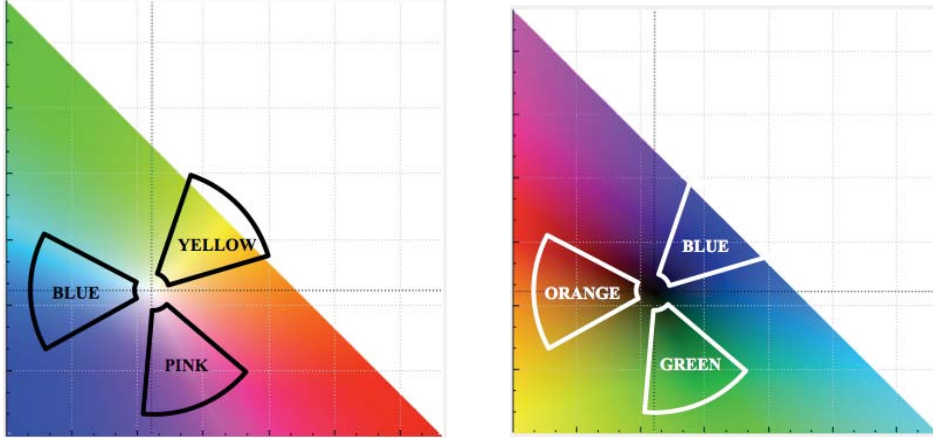


Figure 4-3: Normalised R/G Colour Space (left) and Normalised C/M Colour Space (right) with areas of strength highlighted

When plotted onto the red/green colour space, above left, it can be noted that the areas of strength where two colours meet, for example green/blue, green/yellow and red/blue. It is hypothesised that the colour space excels in these areas due to the ‘squashed’ nature of these regions. Data representing the colours of blue, yellow and pink plotted in these regions are grouped together tightly, ideal for a classifier based on radius and angle ranges. This paper introduces a colour space complementary to the red/green chromaticity; the cyan/magenta chromaticity, above right.

It is expected that this colour space will excel in orange, darker blue colours and green. The formulae for calculating the normalised cm-chromaticities from RGB values are below, highlighting the complementary nature of this space to the rg-chromaticity:

$$c = \frac{|R - 255|}{|R - 255| + |G - 255| + |B - 255|}$$

$$m = \frac{|G - 255|}{|R - 255| + |G - 255| + |B - 255|}$$

Where R , G and B are values of red, green and blue, respectively.

4.3 Multiple Camera Input Pre-Processing

In order to maximise the dynamic ranges and reduce specular lighting reflections, three cameras are used as inputs to the system. These cameras are to be placed at different angles to the planar surface. Each camera will have an exposure ‘locked’, chosen by an arbitrary exposure setting process, discussed in Section 4.3.3. This process will ensure brighter areas are focused on by one camera, darker areas another, while the final camera will focus on the ‘best overall’.

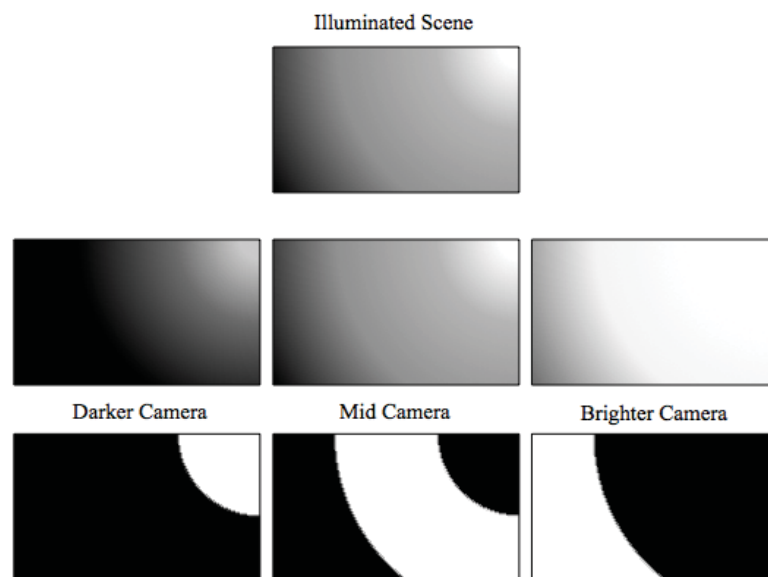


Figure 4-4: Exposure ranges spread across three input cameras

Above, the lower three black and white representations of the input image indicate the regions of strength, white, compared to the regions of ambiguity, depicted in black.

4.3.1 Camera Calibration

Homographic isometric transforms are used to warp the image, correcting the perspective and rendering the input images the same for each camera.

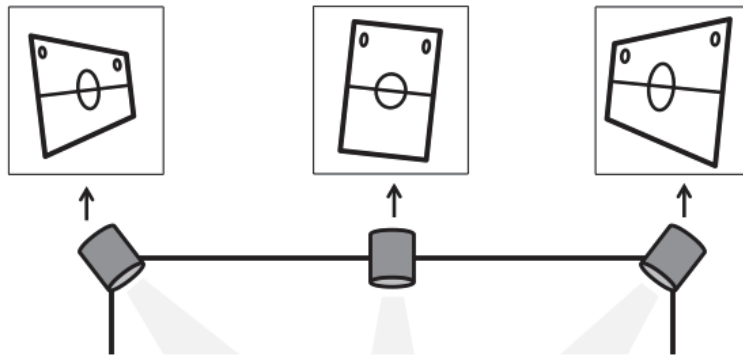


Figure 4-5: Examples of perspective-skewed images from each input camera

Each camera image is then scaled to the same aspect ratio; as isometric transforms may result a slightly wider or longer image. The perspective adjustments are to be made by the user, manually selecting four corners on the camera input image. A camera designated as the ‘main’ input will be chosen to base scaling and isometric distortions from. Ideally the camera that has the least correction required, resulting in the least corrections possible.

Aspect ratio scaling is to be performed by inputting the expected dimensions of the focus surface. The main input image is then used to scale the length and width of each of the remaining input images.

After image perspective and aspect ratio correction, each image may not be aligned with the precision required for comparison. This becomes a problem when identifying a particular pixel between two cameras. Another process of alignment is to be run where a central circle, found on all three input images is automatically identified and position calculated. This position is saved as an alignment-adjust value, further creating a comparable image for each input. The effectiveness of this process will be visible by ‘blending’ each camera image and noting any overlap of the central circle, seen in Section 5.1.1.2.

4.3.2 Scenario and Lighting Capture

Using three cameras, the system is expected to identify colours across varying lighting conditions. In order to identify these differing conditions, the terminology of ‘lighting scenarios’ is used. A single scenario is denoted as a specific layout of light projecting onto the focus surface. These scenes have been sorted into three distinct categories, with varying sub-categories further denoting one of nine conditions;

Even Lighting – A scene where lighting is evenly distributed across the field.

Even lighting may be bright, medium or dim in brightness.

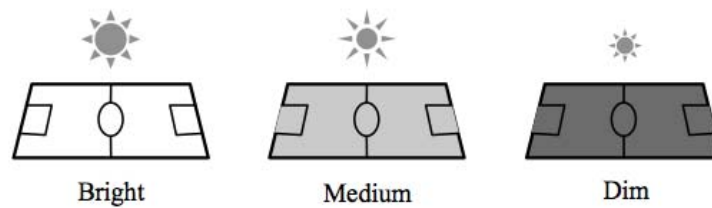


Figure 4-6: Even Lighting Scenario examples

Uneven Lighting – A scene where lighting is darker towards one side of the field.

Uneven lighting may be slight, moderate or severe in dynamic range.

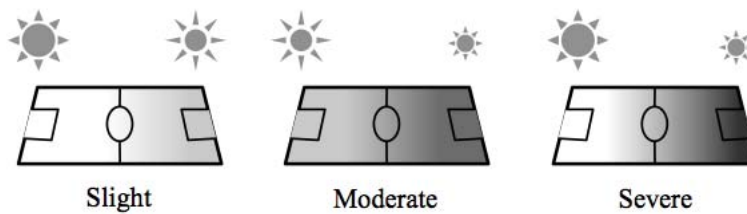


Figure 4-7: Uneven Lighting Scenario examples

Harsh Lighting – A scene where a bright halogen spotlight is used to shine directly onto the playing field.

Harsh lighting may be slight, moderate or severe in dynamic range.

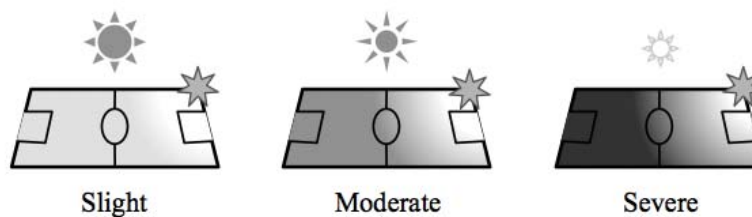


Figure 4-8: Harsh Lighting Scenario examples

The three different categories, coupled with known-areas of the colour patches provide the system with training data to create the pie slice classifiers.

The three input frames from each camera, locked at their unique exposures, are then saved as a single ‘scenario’. Once a single scenario has been saved, the lighting conditions may be changed. The exact illumination values will be measured in Section 5.1.3.1, highlighting each scenes’ illumination variance.

The variation of lighting conditions is important, as turning off a particular set of lights or moving the halogen spotlight may create a different hue for a known-colour. This can be both a positive or negative addition to sample data, depending on the classification technique. At this stage of data processing, the three input image frames are saved as Portable Network Graphics (PNG) files, retaining as much data as possible.

4.3.3 Exposure Selection Process

Crucial to the success of the system, each camera must have an exclusive exposure setting. This setting needs to provide a unique input image, ensuring each camera focuses on a different lighting range. The three exposures are to be determined by stepping through each cameras’ exposure value, recording an overall ‘score’ for each.

The scoring function for each exposure value, for each camera, is calculated as follows:

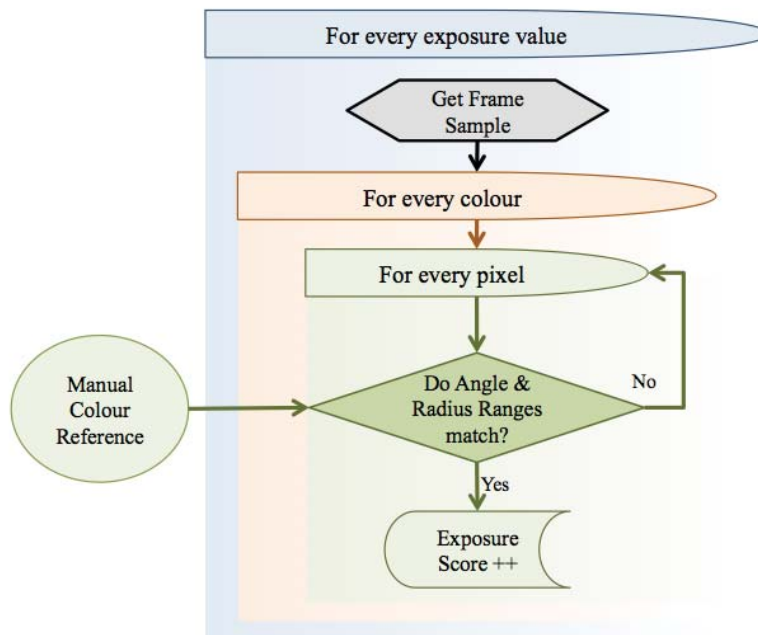


Figure 4-9: MIDECCS Exposure Selection Process

The externally provided manual colour reference is set before the training process begins. Samples of the chosen colours are to be stored as pie slice angle and radius values, selected from an on-screen image by the user. This sets a baseline hue for each classifier and is one of the few manual inputs required. The manual colour reference can be chosen on-screen from any camera. Adding multiple colour references may assist in training, however there are tolerances set so the exact values are not required - see the example in Section 5.1.2.3.

The exposure selection process is slightly different for each camera, as scoring must take into account the pixels previously ‘focused’ on.

The algorithms below explain the differences in the camera exposure selection process:

Algorithm 4-1: Camera A Exposure Selection (Best Overall)

For each $e \leftarrow$ every exposure value from minimum to maximum
 Fetch an input image frame
 Identify colour matches in ‘known areas’ of the input image
 Calculate a score based on the number of matches / total possible matches
 Store the score for this exposure
Lock Camera A exposure at ‘best’ scored exposure
Save ‘mask’ where Camera A has correctly identified a colour

Algorithm 4-2: Camera B Exposure Selection (Darker Exposure, Focuses on Brighter Areas)

For each $e \leftarrow$ every exposure value from Camera A exposure to maximum
 Fetch an input image frame
 Apply ‘mask’ to input image, ignoring pixels already classified by Cam A
 Identify colour matches in ‘known areas’ of the input image
 Calculate score based on the number of matches / total possible matches
 Store the score for this exposure
Lock Camera B exposure at ‘best’ scored exposure

Algorithm 4-3: Camera C Exposure Selection (Brighter Exposure, Focuses on Darker Areas)

For each $e \leftarrow$ every exposure value from Camera A exposure to minimum
 Fetch an input image frame
 Apply ‘mask’ to input image, ignoring pixels already classified by Cam A
 Identify colour matches in ‘known areas’ of the input image
 Calculate score based on the number of matches / total possible matches
 Store the score for this exposure
Lock Camera C exposure at ‘best’ scored exposure

Each of the exposure setting algorithms require the camera image and colour patch locations to generate the exposure score, while Cameras B and C also require the results from Camera A in order to ‘ignore’ the currently ‘focused’ areas.

False positives were not counted in this process as the search hyperspace in the colour space is already so limited, false positives would be minimal and not likely to affect the camera exposure setting process.

The exposure selection process is to be run during the ‘worst case’ lighting conditions. The goal of providing the greatest range of sample exposures is important so a wide range of colour samples are taken into account.

4.4 Sampling Training and Test Data

Generating usable data from the scene PNG files involves conversion to a pie slice angle and radius value. This process must be run twice for each input image pixel - once for each colour space.

At this stage, both colour spaces are simultaneously processed before a weighted score is calculated to determine the best colour space for each colour, for each camera.




		
Camera A: Aligned Correctly, Only sampling patch	Camera B: Misaligned, Sampling background	Camera C: Misaligned, Sampling background

Figure 4-10: Incorrect sampling of known-areas, without margins

Due to camera alignment, highlighting a certain rectangular area at position (x,y) , with size (w,h) in one input image may not necessarily match the exact coordinates in another image.

The solution implemented in this research is to ignore pixel values that are within a certain ‘margin’ of the chosen area. This ensures that the background is not provided as training data, as well as ignoring the edges of the patch that may reflect different colours due to lower resolution cameras.

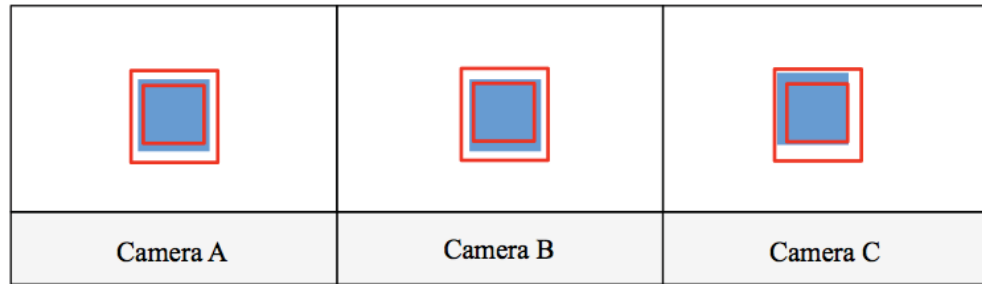


Figure 4-11: Correct sampling of known-areas, with margins

The above example, showing two red outlines around each patch, indicates a margin inside which pixels are ignored both for statistic gathering and classification. In the MIDECC system implementation, a margin value of 6 pixels was used to successfully filter patch edges for data set gathering.

4.4.1 Conversion to Pie Slice

Converting large numbers of RGB values to respective pie slice angle and radius values needs to be as computationally efficient as possible. The largest numbers of computations, processed during exposure selection, are processed using lookup tables (LUTs). The LUTs will ease processing for each pixel in the input images, being used roughly 480,000 times for each exposure setting, for each camera. The tables are pre-generated, holding all 16.5 million possible red, green and blue values with corresponding double precision angle and radius outputs.

The drawback of using such large lookup tables is the loading time at the start of each runtime – approximately 15 seconds for each colour space. Due to this limitation, once the exposure processing has been completed, RGB to pie slice conversion is done ‘on-demand’ for each colour. This is fairly quick due to the limited number of colour conversions required, compared to the whole image processing of the exposure algorithms. RGB values are converted to r/g and c/m colour spaces using equations discussed in Sections 3.2.2 and 4.2 respectively.

4.4.2 Data Collection and Storage

Data is stored separately for comparison classifiers, namely Comma Separated Values (CSV) files to be loaded into Waikato Environment for Knowledge Analysis (WEKA).

Given 14 different lighting scenes, each with 11 different sample areas of roughly 200 pixels in size, the following number of samples are collected for each class:

Table 4-1: Training and test set class sizes

Class	Number of Training and Test Samples
Orange	81,116
Light Blue	85,792
Red	91,378
Pink	91,056
Dark Blue	92,106
Light Green	89,152
Yellow	86,492
Dark Green	88,942
Background	5,740,504

Note that the background samples are only to be used by the MIDECC system during testing, as the training process only requires the known-areas and minimal manual colour references.

The CSV files consist of twelve ‘dimensions’ of data, ensuring the comparison classifiers are ‘fed’ as much data as possible. Each line of data consists of angle (Ang) and radius (Rad) values for each of the three cameras, in both colour spaces.

Table 4-2: Example of CSV file layout for WEKA Classifier training and testing

Camera A				Camera B				Camera C				Expected Value
R/G		C/M		R/G		C/M		R/G		C/M		
Ang	Rad	Ang	Rad	Ang	Rad	Ang	Rad	Ang	Rad	Ang	Rad	

An additional 14 lighting scenarios have been captured with slight variances, for example consecutive frames or slight halogen lamp movements. This set is to be used as a test set.

4.5 Building Classifiers

The following sections in this chapter relate to the unique system proposed by this research. The evaluation of pie slice data to construct the classifier is directly linked to the pie slice and colour space chosen in Section 4.5.2, while classifier construction in the next section is last to consider both colour spaces.

4.5.1 Pie Slice Classifier Construction

Once the approximately 90,000 data points per colour have been converted to angle and radius values, the points are plotted into the colour space.

The angle and radius ranges are then calculated, along with the mean values and standard deviations. These values are then used to assume a normal distribution, used later to calculate the probability or ‘confidence’ of each camera’s match.

Individual colour classifiers are created by adding and subtracting a single standard deviation from the mean angles and radii. By using a spread of one standard deviation, it is expected to cover approximately 68% of the sampled colour data, excluding extreme outliers, such as almost-black or almost-white samples. It is important to exclude extreme values as widening the pie slice classifier both by angle (hue) or radius (saturation) would result in a significant increase in false positive matches.

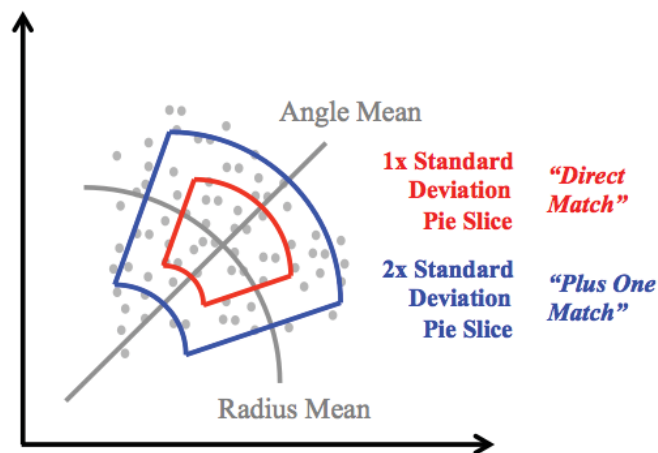


Figure 4-12: Pie Slice classifier, indicating the two priority regions

The classifier matching system includes standard deviation based priority, discussed in Section 4.5.3, further widening the classifier as a ‘last resort’ to match a colour.

Classifiers are individually stored in standard text files. This enables manual checks and editing if required. For a system with three cameras, identifying eight colours, a total of 24 classifiers are generated. For each colour, only the colour space identified in the next section is used to save the classifier.

4.5.2 Colour Space Selection Process

Up until this point, all colours have been processed in parallel using both colour spaces. The next stage of processing selects the best colour space based on a range of evaluation criteria. The classifier is then saved in the best colour space – no further processing is run for that colour in the worse-off colour space.

Colour spaces are compared by the following criteria:

Maximising True Positives; Score = Score + 0.40

Maximise the matches that have been correctly found in the known-areas for this colour.

Minimising False Positives; Score = Score + 0.35

Minimise the matches that have been incorrectly found outside any known-areas for this colour. This is a simple comparison between the two colour space percentages.

Minimise Angle Range; Score = Score + 0.15

Minimise the difference of the maximum and minimum angles for this colour. Angle and radius ranges are calculated by subtracting the minimum value from the maximum value.

Minimise Radius Range; Score = Score + 0.05

Minimise the difference of the maximum and minimum radii for this colour.

Minimise Grey False Negatives; Score = Score + 0.05

Minimise the number of matches that occur less than the minimum radius value. These values are closer to the ambiguous ‘grey area’ of the colour space, discussed in Section 2.2.2.

Once these scores have been added, the colour space with the higher score will be selected for the final classifier for this colour. The scores were chosen based on the criteria’s ability to introduce false negatives or reduce the number of true positives. Manual adjustment of these scores during experimental development has ensured the correct colour space is chosen.

In the event that a tie between colour spaces occurs, the cyan/magenta colour space is selected. This is due to experimental results of the cyan/magenta colour space matching or exceeding the performance of the red/green colour space in most tests.

4.5.3 Standard Deviation Priority during Classification

Discussed previously, each colour classifier is based on the ranges of a single standard deviation from each angle and radius mean. As the data is assumed to be a simple normal distribution, this would expect to cover approximately 68 percent of cases for that exposure.

Adding another standard deviation either side of the range, effectively making the classifier ± 2 standard deviations either side of the mean would result in approximately 95 percent of the data points being covered.

In practice, expecting 95 percent of the data points is difficult due to the layout of each colour space, however this extra standard deviation is seen as a ‘last resort’, only being added if no other classifiers are matched.

Terminology of a ‘direct match’ is used if the classifier has not been extended, matching within one standard deviation, while it is said to be a ‘plus one’ match if an additional standard deviation has been added.

The following algorithm explains the classification for each colour, with respect to the standard deviation priority levels:

Algorithm 4-4: Matching a classifier with multiple priority levels

```

For each  $c \leftarrow$  every classifier
    Check for a direct match with classifier  $c$ 
    Check for a plus one match with classifier  $c$ 
If one or more direct matches Then
    Find best colour (see Section 4.5.4)
Else If one or more plus one matches Then
    Find best colour (see Section 4.5.4)
Else no match for this pixel
  
```

Examples below explain how the prioritising system works with regard to the two classifier ‘match types’:


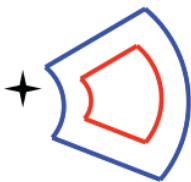

		
Camera A: Dark Blue, Direct Match	Camera B: No Match, Too Dark	Camera C: Light Blue, Plus One Match

Figure 4-13: Example of multiple classifiers matching Dark Blue

Seen above, Camera A and Camera C have both matched with Blue classifiers, however Camera A matched Dark Blue, while Camera B matched Light Blue.

As Camera A has matched its colour within the red ‘direct match’ (single standard deviation) area, this camera takes priority over Camera C. The final output would be Dark Blue.




		
Camera A: Orange, Plus One Match	Camera B: Yellow, Direct Match	Camera C: Orange, Plus One Match

Figure 4-14: Example of multiple classifiers matching Yellow

Despite two cameras matching the same colour, priority is given to the colour which has a direct match, this being Camera B, outputting Yellow.




		
Camera A: Dark Blue, Plus One Match	Camera B: Dark Green, Plus One Match	Camera C: Dark Blue, Plus One Match

Figure 4-15: Example of multiple classifiers matching, with no priority

This example does not have a direct match, so there is no standard deviation ordering required. It is notable that all cameras have matched contradictory colours, so further processing is required to decide which colour is correct. The normal density function, discussed in the next section calculates the probability of each classifier, picking the most probable.

4.5.4 Handling Multiple Classifications

There are occasions when the system has classified multiple colours for the same pixel. This may be due to overlapping classifiers or particularly difficult lighting conditions, where two cameras classify different colours. These conflicts are resolved by comparing the normal density function of both the angle and radii of conflicting colours.

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The normal density function above, is run for both the radius and angle of each classifier match. These values, when added together, create a value which increases the closer to the mean point of each pie slice. It would be expected that a data point aligned perfectly in the centre of the pie slice classifier would output a normal density value close to 1, while a further away point would output a number closer to 0.

The output from this function is treated as a confidence level, prioritising one particular colour above others when multiple matches occur. In manual calculation tests, multiple matches between similar colours, such as light blue and dark blue, were resolved successfully with the correct colour being classified.

4.6 Background Removal

Discussed in multiple sections, the 'grey area' of the r/g colour space highlights the small area around the origin where colour classification is unreliable.

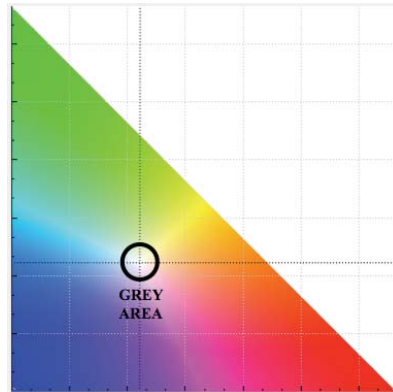


Figure 4-16: Grey Area, highlighted in the R/G colour space

The colours which would be classified in this region would include grey, white and black. This fact is to be used for background removal, ignoring pixels if they are below a certain threshold.

A small amount of fine tuning of the constant filtering value is to be expected - setting the value too low may result in the background not being removed, while setting the value too high may result in the filtering of valuable colour information. The ideal value for background removal will result in no or little loss of classification reliability.

4.7 Evaluating Classifiers

Evaluating the effectiveness of a classifier can be done in multiple ways:

- Training the system without a 'background' class, effectively limiting the system to identifying the known colours only.
This would result in a specialised system which cannot identify background pixels, attempting to classify every sample as a known colour (including a background pixel). This is a valid assessment method if the background was filtered out of an image during a pre-processing phase for example.
- Training the system with a background class, evaluating both the classification of known areas (true positives) and misclassifications of known areas and classifications outside known areas (false positives).

This results in a general classification system, allowing any colour, background or not, to be classified as either a valid classified colour or part of the background.

The MIDECC system has the advantage of being able to train from the known-colours only, without the need for additional background samples. This is explained and compared further to the requirements of the WEKA standard classifiers in the next section.

4.7.1 Classifier Evaluation Methods

It has been decided to compare the MIDECC system with both methods described above, training once without background information before being tested to classify all lighting scenarios with and without background data.

The WEKA classifiers will be trained and evaluated twice, once without background data or ‘known-areas only’, then again with background data.

Table 4-3: Data Collected for Training and Testing

Process	Data to Collect
Training, Known-Areas Only	Colour patches only from the Training Set
Testing, Known-Areas Only	Colour patches only from the Test Set
Training, Background Included	Colour Patches and Background, ignoring margins from the Training Set
Testing, Background Included	Colour Patches and Background, ignoring margins from the Test Set

4.7.2 Sampling Test Data

Discussed in Section 4.5.1, test data samples have been collected and stored as CSV files. Test data consists of the same lighting scenes as training, however input image frames have been refreshed and the lighting subtly adjusted. This provides different data for the system to test itself, yet still within the same scenarios as explained in Section 4.3.2. This is the same data that will be sent to the Waikato Environment for Knowledge Analysis (WEKA) program.

4.8 Summary

Proposed is a new concept to create classifiers that are based on multiple lighting scenarios.

The exposure setting process for each camera ensures different cameras focus on different areas with minimal overlap. The use of different lighting scenarios ensure the pie slice colour space descriptors generate a spread of data points, while assuming a normally distributed dataset ensures a simple normal density function may prioritise multiple matches if required.

In the next chapter, the MIDECC system is implemented and compared against common machine learning algorithms.

Chapter 5

5. Experimental Setup

Experiments were performed using three cameras mounted above a robot soccer playing field in a computer laboratory with controlled lighting. Eight target colours spread across eleven regions are assessed at a capture resolution of 800 x 600 (WVGA). The eleven regions were strategically positioned across the field to ensure a spread of differing illumination conditions. The lighting in the room can be controlled to a repeatable level, matching the discussed lighting scenarios in the previous chapters.

5.1 Environment Setup

The environment choice for testing the proposed system is significant as its' results will help determine the scalability and uniformity of results. Consumer electronics were used where appropriate to ensure the results weren't subject to specialised hardware.

5.1.1 Cameras

Three Logitech QuickCam Pro 9000 web cameras were used as inputs for the system. The auto focus feature was enabled on power up only, to ensure a focused input image. Auto exposure was disabled, as the exposure setting process manually sets this value. The cameras were run at WVGA (800x600) resolution as opposed to their native 2-megapixel rating as three cameras at WVGA resolution met the limit of the Universal Serial Bus (USB) connection to the processing computer. Commands are sent to the cameras via the 'Video 4 Linux 2' (v4l2) command line interface, enabling the system to command the cameras directly.



Figure 5-1: Logitech Camera locations, emphasis added

5.1.1.1 Locations

The three cameras were located approximately 2.5 meters above a robot soccer field, centred length-wise and spread along the fields' width. The two cameras near each edge of the field are angled towards the centre at approximately 25 degrees. Mentioned previously, this layout is used to minimise any reflective properties of the playing field reflecting the harsh halogen light back into the camera lenses.

5.1.1.2 Transformations

Removing the perspective angle and aligning the three input images is crucial for pixel-to-pixel comparisons. If the images were not aligned, one pixel, which is at position (150,250) to one camera, may be at location (154,248) on another for example. This can result in many conflicts where one camera is at an edge of a colour patch, while another is on the black background. In the worst case scenario, this would severely negatively affect training and exposure adjustments.

The Open Computer Vision (OpenCV) library is used for image transformation, as it is optimised at performing image warping and homographic isomorphic adjustments.

Firstly, the edges of the playing field are set manually by selection on the camera input image. This sets four 'perspective' points from which the homography matrix may be calculated using the `findHomography()` function. This function returns an adjustment matrix, which removes the image perspective when passed to the OpenCV `warpPerspective()` function.

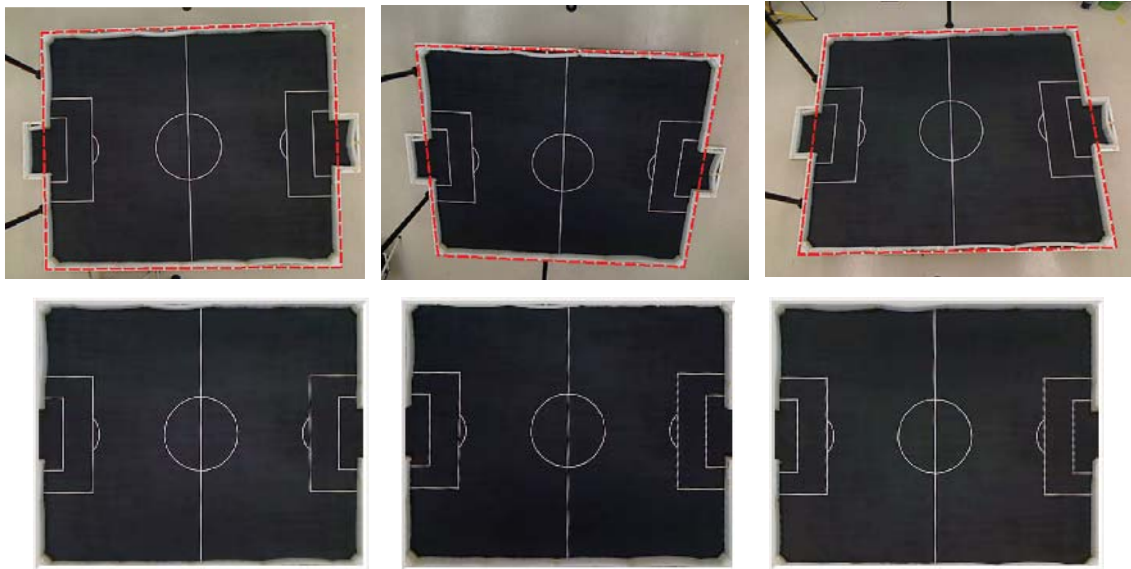


Figure 5-2: Camera Raw Input images with red outline around focus plane (top), same images with pre-processing completed (bottom)

Secondly, the images are stretched slightly to match a standard aspect ratio. This ensures that once the homography warping is complete, squares remain as squares while the central playing field circle, used in the next processing step, also remains circular. The aspect ratio is calculated by the real-life playing field measurements, using the camera in the centre of the playing field as a reference.

Lastly, the three images are shifted on the X and Y axis to align the centre circle on the playing field. This process uses the `houghCircles()` function to find the central white circle on the playing field. The location differences are then calculated and saved with each camera calibration file, resulting in faster future processing.

Table 5-1: Camera Alignment values

Camera A	Reference Camera
Camera B	+2 on X-Axis, +9 on Y-Axis
Camera C	+3 on X-Axis, -5 on Y-Axis



Figure 5-3: Camera Alignment testing output

The alignment test image above, appears as a single circle cropped from the centre of the input images. This image actually has 33% opacity from each camera, blended together to give a preview of the alignment results. An image with no overlap or visible ‘ghosting’ around the white central lines means each camera is successfully aligned.

5.1.2 Known-Area Colour Patches

Eight colours are to be detected in eleven different areas on the playing field, totalling 88 different patches in one frame. The locations of the eleven patches corresponding to each colour are saved with the program, enabling the system to keep track of expected colour locations. The eight colours are reproduced using coloured matte paper. These squares of paper each measure approximately 80 by 80 mm in size, turning into roughly 20 x 20 pixel rectangles on the input camera image.



Figure 5-4: Sample of known-area colour patches on the focusing plane

Mentioned in Section 4.4, 6 pixel wide margins are used to assist in camera alignment and minimise false data collection. Figure 5-5 highlights the margin filtering between multiple cameras for Area 2, Dark Blue in the ‘Even Bright’ training scenario.



Figure 5-5: Area 2 highlighting the importance of the margin value for each camera

5.1.2.1 Colour Choices

The following table shows the colours that are used in this research experiment. Colours in the table have been picked directly from Camera A (best overall exposure), ‘Even Bright’ training lighting scenario.

Table 5-2: Colours sampled from pre-processed images

<i>Red</i>	<i>Orange</i>	<i>Yellow</i>	<i>Pink</i>	<i>Light Blue</i>	<i>Dark Blue</i>	<i>Light Green</i>	<i>Dark Green</i>

These colours have been chosen both to test the range of hues available while challenging the system with similar colours. Light and dark blue are difficult to distinguish in certain lighting conditions, while dark green may turn into light green in bright conditions. It is a goal for the system to continue working reliably under varying illuminations.

5.1.2.2 Patch Locations

The locations of the eleven groups of colours have been chosen to offer a wide range of illuminations per scenario. This is especially noticeable during the ‘severe’ lighting scenarios, demonstrated below.



Figure 5-6: Patch locations throughout the focusing plane, viewed through multiple cameras

5.1.2.3 Manual Colour Entry

During the exposure setting process, the system assigns a score to each exposure by searching through known areas with a basic pie slice classifier. This classifier is generated manually by clicking on examples of the colour on the input images. These ‘hand trained’ classifiers are only used for the exposure selection process, after which the automatically generated versions are trained. Figure 5-7 shows samples of the manually trained classifier for dark green. The outer ‘slice’ features an added tolerance, matching a greater range of colours. While the lesser the tolerance the better, the values seen below allow the known-areas to be found without introducing further noise or false positives. It is also worth noting that these manually trained classifiers are only specialised at the single lighting scenario they are created in – further training is required to create a general illumination independent classifier.

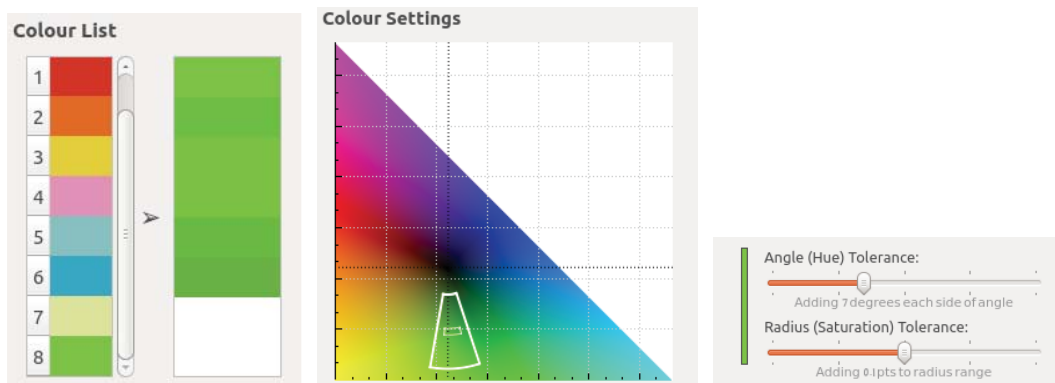


Figure 5-7: Manual colour entry samples (left), the manually classified pie slice (centre), with tolerance adjustments (right)

Indicating the manual colour entry test points for dark green, the six entered colours on the left side of Figure 5-7 generate the smaller pie slice in the centre image.

The tolerance settings on the right are adding 7 degrees either side of the manually entered colours, while adding 0.1 points to the radius range, generating the thicker-lined pie slice seen in the c/m colour space. It is this pie slice which is used as the reference colour when the automatic exposure process is running.

5.1.3 Lighting

Controlled lighting is essential for this research, as small changes in illumination may result in previously measured lighting levels becoming incomparable for training.

The robotics computer laboratory is able to provide many different lighting combinations due to the multitude of light sources specially built above the robot soccer field.

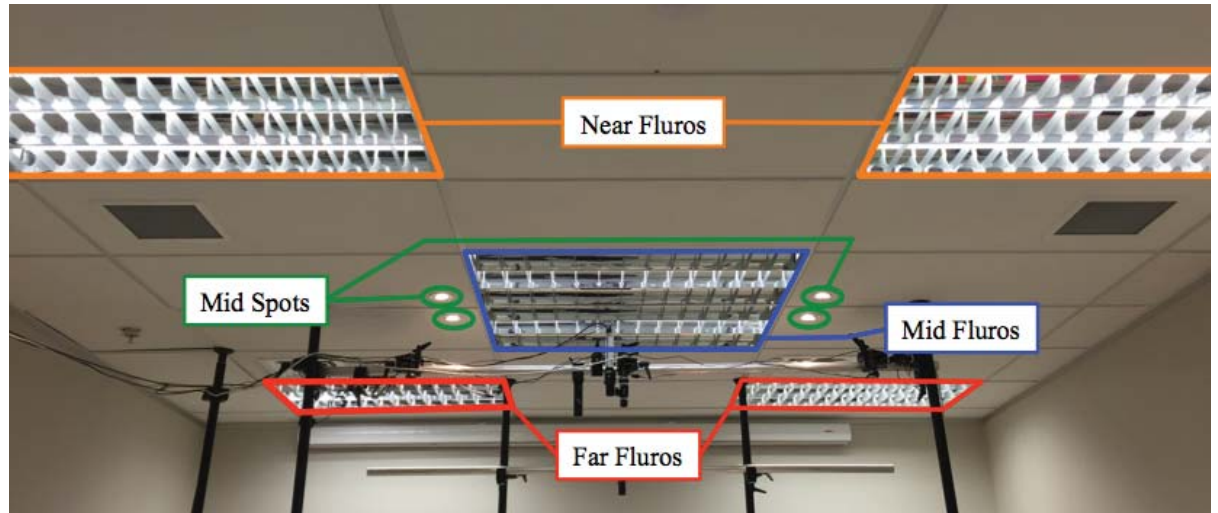


Figure 5-8: Room lighting locations and terminology

Each corner of the playing field has a pair of overhead in-ceiling fluorescent lights, designated ‘Close Fluors’ and ‘Far Fluors’. A third cluster of florescent lights are placed over the centre of the soccer field, named ‘Mid Fluors’.

Four halogen spotlights, spaced around the Mid Fluors are given the name ‘Mid Spots’, while a portable halogen floodlight provides harsh lighting, to be moved to the corner and edge of the field when required.

5.1.3.1 Scenarios

Each combination of lighting results in a different lighting ‘scenario’. This creates a unique set of data points for training. Despite training data being combined, it is important to be able to identify problematic scenes and compare results during classifier evaluation.

Table 5-3 describes each scenario, with the illumination values measured at each corner and once in the centre. Illumination values were measured with a Konica Minolta CL-200A portable Chroma meter, depicted Figure 5-9.



Figure 5-9: Measuring illumination in the centre of the plane (left) and in corners (centre), corner terminology (right)

Table 5-3: Lighting Scenarios

Scenario Name	Lighting Setup	Expected Illumination	Corner 1	Corner 2	Corner 3	Corner 4	Centre
Even Lighting							
Bright			1,378 lux	1,383 lux	1,376 lux	1,382 lux	1,498 lux
Medium			884 lux	901 lux	892 lux	876 lux	751 lux
Dim			364 lux	300 lux	293 lux	372 lux	581 lux
Uneven Lighting							
Slight			865 lux	439 lux	415 lux	832 lux	685 lux
Moderate 1			345 lux	696 lux	725 lux	327 lux	568 lux
Moderate 2			669 lux	263 lux	250 lux	689 lux	603 lux
Severe 1			210 lux	601 lux	667 lux	246 lux	382 lux
Severe 2			529 lux	206 lux	150 lux	458 lux	221 lux
Harsh Lighting							
Slight 1			1,311 lux	11,230 lux	1,266 lux	1,216 lux	1,694 lux
Slight 2			1,171 lux	1,646 lux	1,657 lux	1,197 lux	1,119 lux
Moderate 1			607 lux	10,680 lux	448 lux	515 lux	1,225 lux
Moderate 2			514 lux	1,124 lux	1,097 lux	494 lux	1,375 lux
Severe 1			220 lux	10,170 lux	175 lux	129 lux	226 lux
Severe 2			176 lux	860 lux	830 lux	180 lux	873 lux

5.2 Exposure Selection

Each of the Logitech cameras has been programmed to adjust to 43 different exposure times, ranging from 1/80 (very bright image, able to focus on darker areas) to 1/18000 (very dark image, able to focus on brighter areas). The exposure selection process discussed in Section 4.3.3 is followed, assessing pixels in the known-areas for each colour.

For the results obtained in this research, exposure selection was run with Scenario 'Harsh Severe 1'. This successfully spread the camera exposures across bright, medium and dark regions. For the remainder of this research and classification results, the three cameras have exposures locked as follows:

Table 5-4: Locked exposure settings for each camera

Camera	Visual Identifier	Strength	Exposure Value
Camera A	Mid-Exposure Image	Mid-Exposed Areas	1/320
Camera B	Darker Image	Brighter Areas	1/1120
Camera C	Brighter Image	Darker Areas	1/280



Figure 5-10: Multiple camera inputs highlighting exposure spread for scenario 'Harsh Severe 1'

The process takes approximately 15 seconds to complete, limited by the time taken for the web cameras to stabilise after the exposure adjust command is called. Wait-times are used in the program code to slow the statistic gathering, otherwise processing would complete before a valid exposure reading is taken.

5.3 Lighting Scenario Capture

Once the cameras are locked to the exposures mentioned in Table 5-4, all 14 lighting scenarios were captured to file. This involved recreating the lighting setups listed in Table 5-3, running homography, aspect ratio adjustment and alignment before saving each frame to a Portable Network Graphics (PNG) file. This resulted in 42 PNG files saved for training purposes – three images for each scenario.

An additional 42 PNG files were saved for testing purposes after slight adjustments were made to the lighting conditions.

The Portable Network Graphics (PNG) file format was chosen to save data, as it is a lossless bitmap image format which can be read and written by both Qt development software and OpenCV.

5.4 Building Classifiers

Sampling the known-areas is processed scenario-by-scenario, colour-by-colour. The method is flexible in the selection of which scenarios are used for training, with immediate feedback on true positives, false positives and a visual spatial layout of data points in the colour spaces.

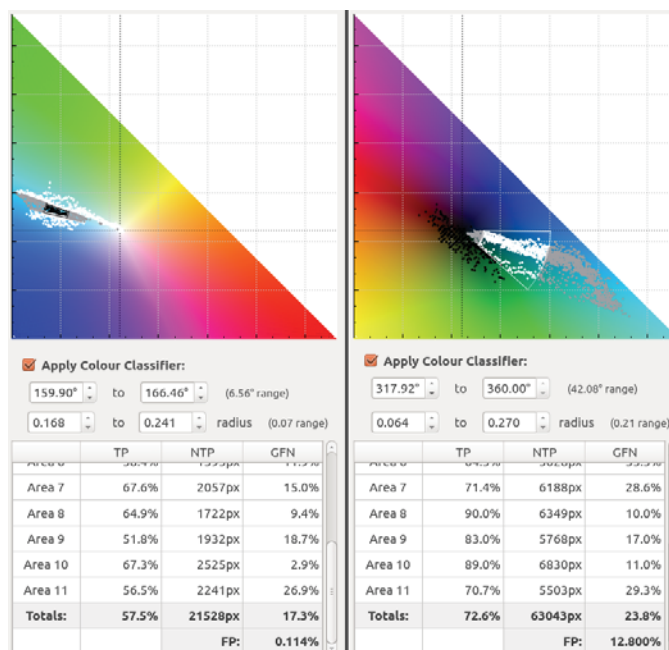


Figure 5-11: MIDECCS Experimental Program outputting classifiers in both colour spaces with corresponding statistics

At this point if a training data sample is below a certain radius threshold, it is discarded. This is the only ‘thinning’ of data that is done throughout the collection and sorting process. The fixed value for both colour spaces ensures that ‘grey area’ data from cameras do not negatively affect the training data. Through manual tuning, any values less than 0.015 in radius are ignored. The below colour space charts outline the difference between ‘grey area filtering’ and standard collection of all data.

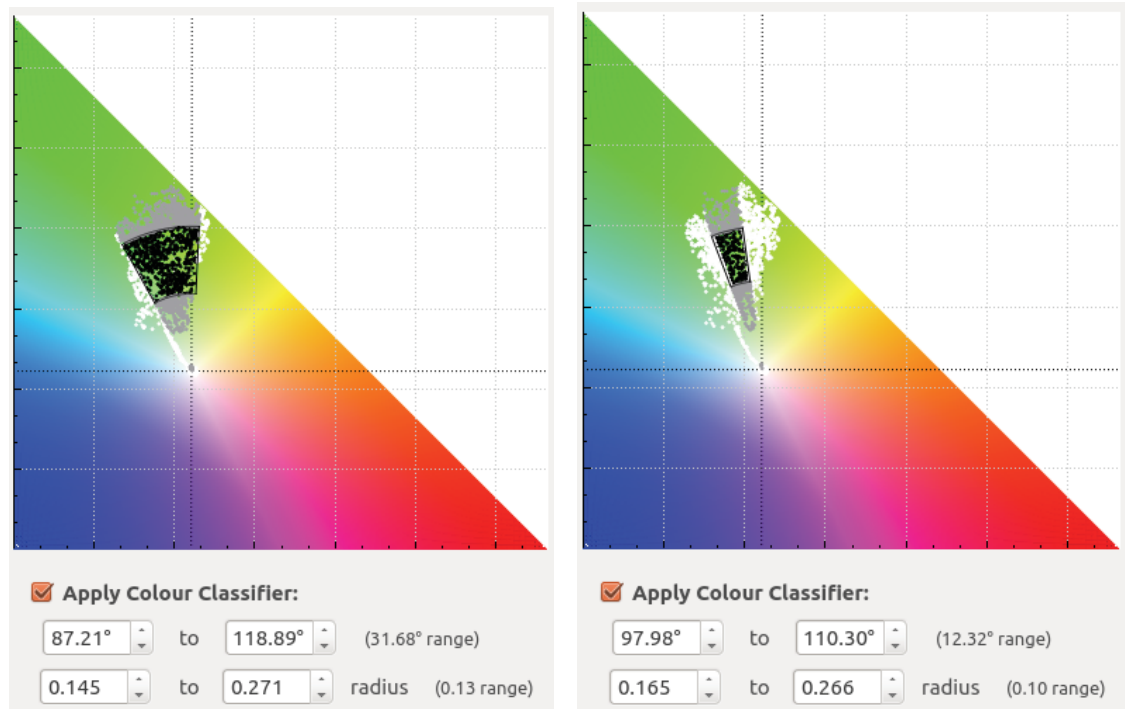


Figure 5-12: Grey Area filtering disabled, producing a wider pie slice classifier (left), Grey Area filtering enabled at 0.015, producing the preferred classifier (right)

Based on the spatial layout and pie slice classifier ranges alone, often it is visually clear which colour space should be used for each classifier, due to one colour spaces’ data points being scattered or clumped together. The ideal colour space should have the data points grouped within a small area, not too close to the ‘grey area’ centre. At this point, weightings of the true positives compared to the false negatives are to be assessed, as a colour space with a brilliant true positive score should not be used if the corresponding false negative percentage is also high.

This process has been automated using the weighted colour space selection process discussed in Section 4.5.2. Pages 78 to 79 highlight some of the comparative outcomes of this process.

Table 5-5: Colour Space selection example for Camera A, Red

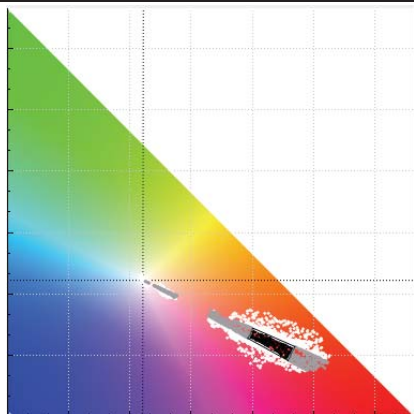
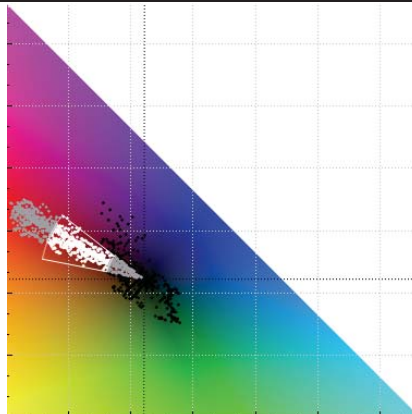
Camera A, Red					
Normalised Red/Green Colour Space			Normalised Cyan/Magenta Colour Space		
					
Pie Slice Angles:	313.12° - 336.5° (5.23° range)	+0.15	Pie Slice Angles:	141.6° - 168.7° (27.08° range)	
Pie Slice Radii:	0.293 - 0.405 (0.11 range)	+0.05	Pie Slice Radii:	0.092 - 0.251 (0.16 range)	
True Positives:	50.9%		True Positives:	65.1%	+0.40
False Positives:	0.519%	+0.35	False Positives:	13.222%	
Grey False Neg:	15.8%	+0.05	Grey False Neg:	32.1%	
Total Red/Green Score:		0.60	Total Cyan/Magenta Score:		0.40
Weighted Colour Space Choice: Normalised Red/Green					

Table 5-6: Colour Space selection example for Camera A, Light Green

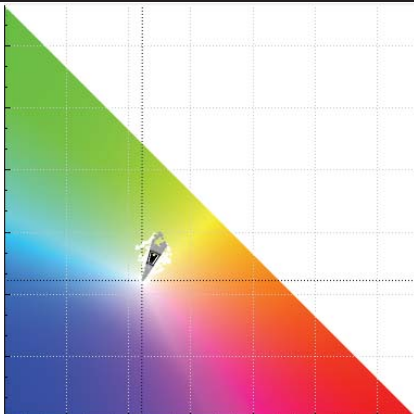
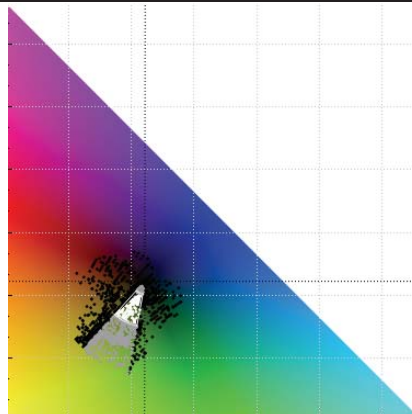
Camera A, Light Green					
Normalised Red/Green Colour Space			Normalised Cyan/Magenta Colour Space		
					
Pie Slice Angles:	56.04° - 74.94° (18.09° range)	+0.15	Pie Slice Angles:	227.7° - 259.5° (31.75° range)	
Pie Slice Radii:	0.044 - 0.075 (0.03 range)	+0.05	Pie Slice Radii:	0.031 - 0.108 (0.08 range)	
True Positives:	42.5%		True Positives:	64.0%	+0.40
False Positives:	1.945%	+0.35	False Positives:	7.604%	
Grey False Neg:	24.5%		Grey False Neg:	19.3%	+0.05
Total Red/Green Score: 0.55			Total Cyan/Magenta Score: 0.45		
Weighted Colour Space Choice: Normalised Red/Green					

Table 5-7: Colour Space selection example for Camera B, Red

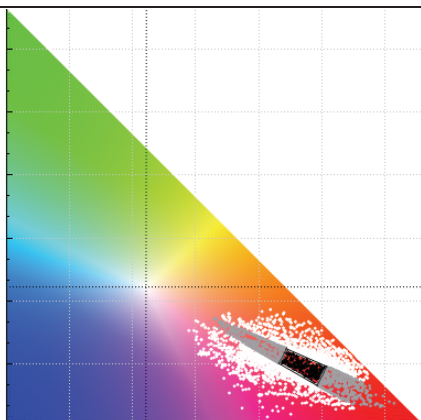
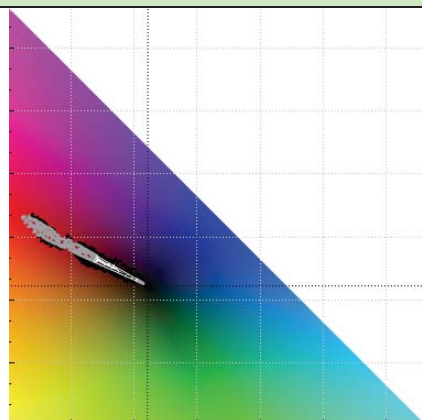
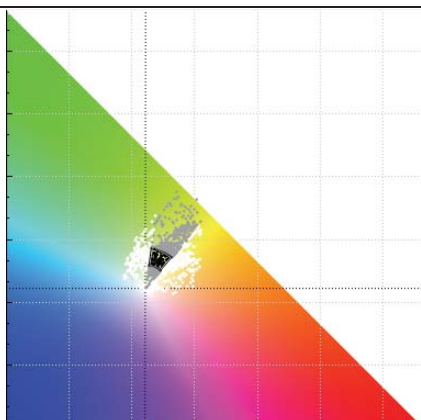
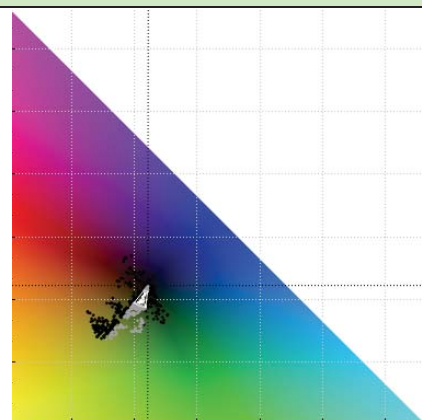
Camera B, Red			
Normalised Red/Green Colour Space		Normalised Cyan/Magenta Colour Space	
			
Pie Slice Angles:	329.99° - 336.20° (6.21° range)	Pie Slice Angles:	149.5° - 155.6° (6.14° range) +0.15
Pie Slice Radii:	0.364 - 0.469 (0.10 range) +0.05	Pie Slice Radii:	0.031 - 0.141 (0.11 range)
True Positives:	53.9%	True Positives:	58.5% +0.40
False Positives:	0.561% +0.35	False Positives:	3.802%
Grey False Neg:	20.5%	Grey False Neg:	15.6% +0.05
Total Red/Green Score: 0.40		Total Cyan/Magenta Score: 0.60	
Weighted Colour Space Choice: Normalised Cyan/Magenta			

Table 5-8: Colour Space selection example for Camera B, Light Green

Camera B, Light Green			
Normalised Red/Green Colour Space		Normalised Cyan/Magenta Colour Space	
			
Pie Slice Angles:	49.71° - 82.41° (32.70° range)	Pie Slice Angles:	231.3° - 262.7° (31.33° range) +0.15
Pie Slice Radii:	0.055 - 0.097 (0.04 range)	Pie Slice Radii:	0.015 - 0.053 (0.04 range)
True Positives:	47.8%	True Positives:	52.8% +0.40
False Positives:	11.153%	False Positives:	5.006% +0.35
Grey False Neg:	22.8%	Grey False Neg:	15.1% +0.05
Total Red/Green Score: 0.00		Total Cyan/Magenta Score: 0.95	
Weighted Colour Space Choice: Normalised Cyan/Magenta			

5.5 Classifier Evaluation

At this stage of the process, each camera input has a preferred colour space for each colour and the pie slice classifier angle and radius have been optimised. The following table lists each classifier, preferred colour space and statistic features.

Table 5-9: Final Classifiers to be used by the MIDECC System

Colour	Colour Space	Angle Range	Angle Mean	Angle Std Dev	Radius Range	Radius Mean	Radius Std Dev
Camera A							
Red	R/G	331.12° - 336.35°	333.73°	2.615°	0.293 - 0.405	0.349	0.056
Orange	C/M	157.03° - 184.33°	170.68°	13.650°	0.106 - 0.249	0.177	0.072
Yellow	R/G	20.54° - 31.75°	26.15°	5.602°	0.155 - 0.23	0.193	0.379
Pink	R/G	316.53° - 325.38°	320.96°	4.427°	0.099 - 0.145	0.122	0.023
Light Blue	R/G	148.52° - 157.81°	153.16°	4.645°	0.066 - 0.103	0.084	0.018
Dark Blue	R/G	159.9° - 166.46°	163.18°	3.280°	0.168 - 0.241	0.205	0.037
Light Green	R/G	56.04° - 74.94°	65.49°	9.448°	0.044 - 0.075	0.060	0.015
Dark Green	R/G	97.98° - 110.3°	104.14°	6.159°	0.165 - 0.266	0.215	0.050
Camera B							
Red	C/M	149.54° - 155.68°	152.61°	3.071°	0.031 - 0.141	0.086	0.055
Orange	C/M	159.41° - 167.11°	163.26°	3.846°	0.041 - 0.141	0.091	0.050
Yellow	C/M	192.39° - 205.99°	199.19°	6.798°	0.025 - 0.115	0.070	0.045
Pink	C/M	135.61° - 146.92°	141.26°	5.652°	0.024 - 0.111	0.067	0.044
Light Blue	R/G	146.8° - 162.28°	154.54°	7.741°	0.080 - 0.133	0.106	0.027
Dark Blue	R/G	160.49° - 171.42°	165.96°	5.467°	0.193 - 0.274	0.233	0.040
Light Green	C/M	231.38° - 262.71°	247.04°	15.665°	0.015 - 0.053	0.034	0.019
Dark Green	C/M	281.54° - 296.4°	288.97°	7.430°	0.023 - 0.085	0.054	0.031
Camera C							
Red	C/M	329.76° - 337.38°	333.57°	3.813°	0.273 - 0.413	0.343	0.070
Orange	R/G	157.11° - 187.04°	172.08°	14.962°	0.188 - 0.257	0.188	0.069
Yellow	R/G	21.12° - 33.64°	27.38°	6.260°	0.145 - 0.233	0.189	0.044
Pink	R/G	315.41° - 325.5°	320.31°	4.898°	0.092 - 0.141	0.117	0.024
Light Blue	R/G	148.81° - 157.76°	153.28°	4.473°	0.064 - 0.101	0.082	0.018
Dark Blue	R/G	158.53° - 165.76°	162.15°	3.617°	0.156 - 0.248	0.202	0.046
Light Green	R/G	56.77° - 78.59°	67.68°	10.910°	0.043 - 0.073	0.058	0.015
Dark Green	R/G	97.58° - 110.48°	104.03°	6.450°	0.158 - 0.264	0.211	0.053

Indicated in the table, each colour now has a preferred colour space to run classification in. It can be noted that Camera B, the camera which focuses on the brighter colour patches, predominantly is associated with the c/m colour space, while Cameras A and C are associated with the r/g colour space. The positioning of each of the colours in these colour spaces are depicted in the diagrams on Page 81.

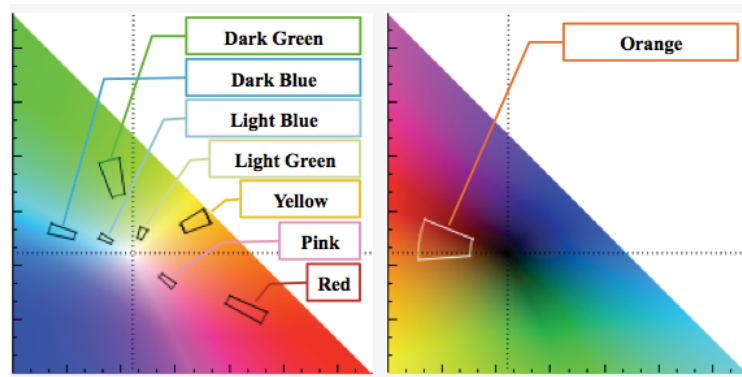


Figure 5-13: Classifier Layout for Camera A

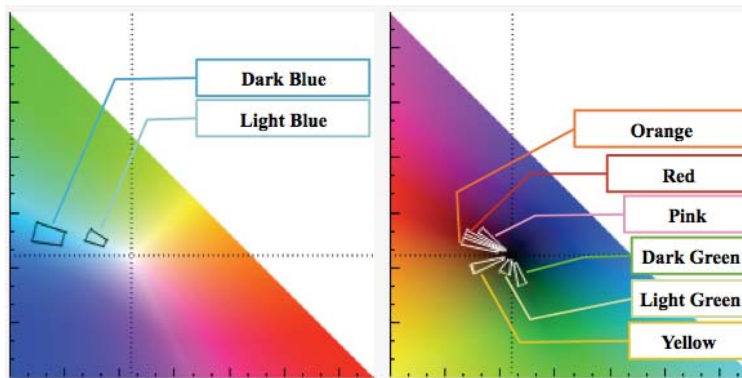


Figure 5-14: Classifier Layout for Camera B

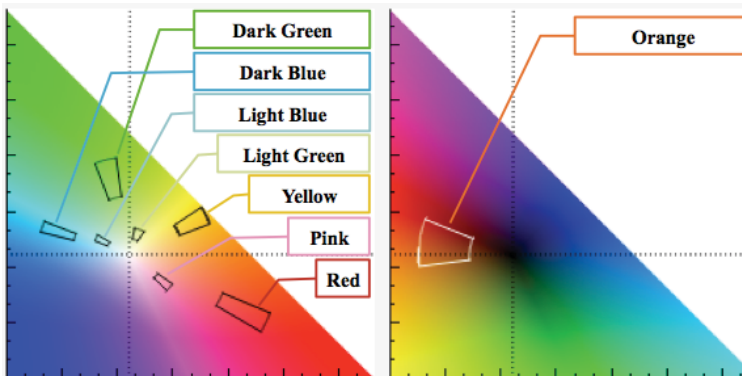


Figure 5-15: Classifier Layout for Camera C

5.6 Background Removal

The background of the output image is filtered to reduce noise and false positives. As discussed previously, a minimum radius value is to be set as a threshold in an attempt to reduce incorrect matches occurring in black, grey or white areas of the image.

This experiment has proven to be very successful at reducing the background noise, with little to no effect on the classification results of known-areas. Results from the MIDECC system were processed using a background removal radius of 0.035, meaning any pixels with a r/g or c/m radius value of less than 0.035 were ignored.



Figure 5-16: MIDECCS Output of 'Even Bright' training scenario with background removal turned off (left half of image) compared to turned on (right half of image)

With background filtering turned off (left half of image), the system classifies many more lighter colours as these are closer to the central 'grey area' of the colour spaces. The system reports 20.66% of the pixels in the output image are noise, while with noise filtering turned on, this number falls to 1.68%.

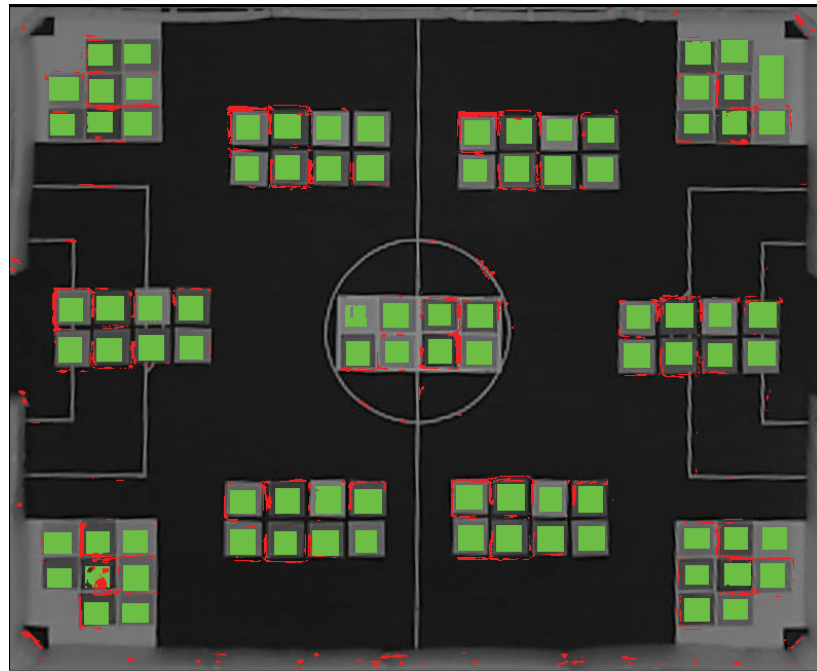


Figure 5-17: MIDECCS Output for 'Even Bright' training scenario with statistic shading

Depicting the 'stats shading' of noise for the 'Even Bright' training scenario, the above image highlights classifications outside known-areas in red, while classifications in known-areas as green. There is small amount of noise along the bottom edge of the image, where the white foam barrier has been classified as orange and yellow (See Figure 5-16 above) and a small amount of noise on the white lines of the playing field. The misclassifications around the edges of the known areas are often identified as the colour orange. Likely coming from Camera A or C, these orange misclassifications may be a result of the large angle range that the system has generated for this classifier. These relatively small amounts of noise are not a concern as setting a 'minimum size' for an expected colour patch would safely ignore these areas.

It is also noted that a small amount of detail may be lost when classifying the lighter colours, visible in the light green (upper left patch) of the central collection of colour patches. Decreasing the minimum radius value to 0.33 fills this patch in, at the expense of adding more noise to the image. The value of 0.35 has been manually adjusted to provide the least amount of classification disruption possible, while removing the maximum amount of background noise.

The graph below show the classification results per colour, both with and without the background noise reduction.

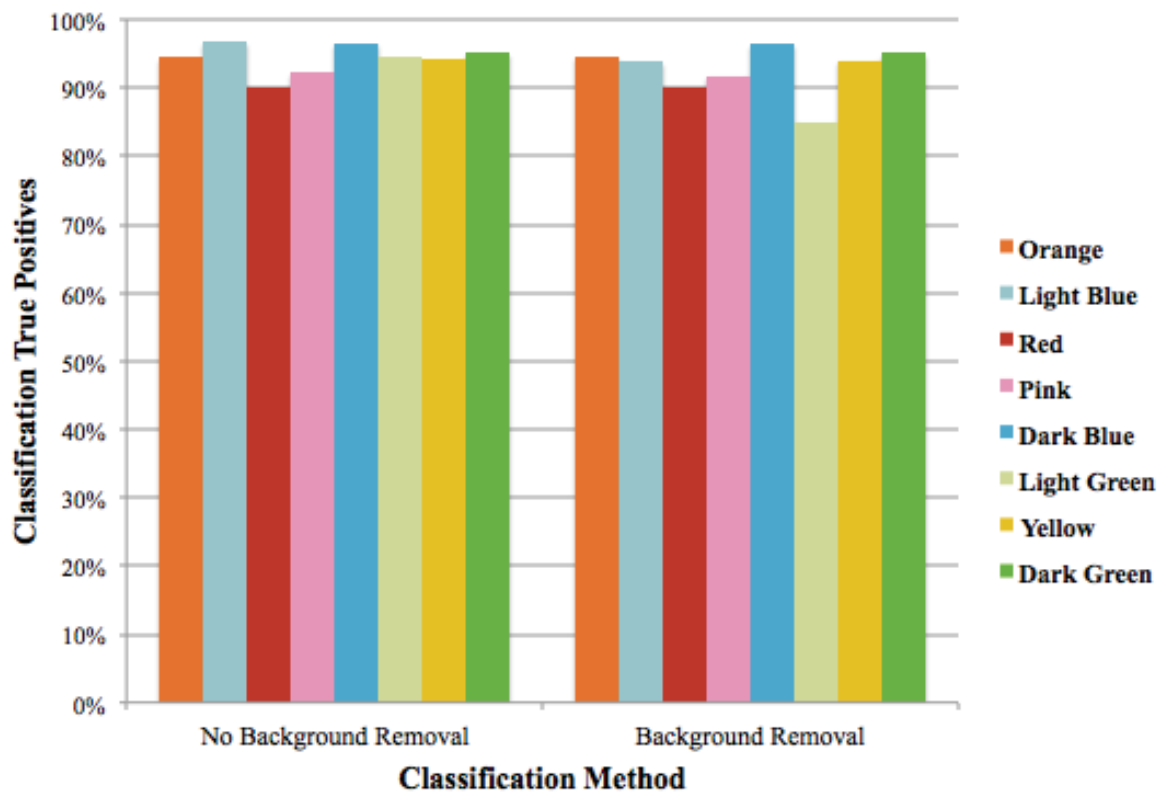


Figure 5-18: Chart indicating minimal difference in true positives with background removal

Chapter 6

6. Experiment Results and Analysis

It is important to note that the following true positive and false positive result outputs from the MIDECC system do not add to 100%. This is due to the heuristic based classification system not matching every single sample provided. If a particular sample does not match any given classifier, it does not return a result, neither classifying a colour or background sample. It is possible to calculate the ‘missing’ or ‘unclassified’ pixels by evaluating the difference between the true positives, false positives and 100%.

6.1 System Results; Known-Areas Only

The MIDECC system was trained with the following number of instances per class:

Table 6-1: Training sample sizes, known-areas only

Class	Number of Samples
Orange	81,116
Light Blue	85,792
Red	91,378
Pink	91,056
Dark Blue	92,106
Light Green	89,152
Yellow	86,492
Dark Green	88,942
	706,034

Identical data was supplied to WEKA and the MIDECC system, generating the following results:

Table 6-2: Classifier training set results, known-areas only

Training Set	True Positives	False Positives
MIDECC System ¹	92.582 %	2.050 %
Naïve Bayes Classifiers	98.478 %	1.523 %
J48 Tree Classifiers	99.985 %	0.012 %
Random Tree Classifiers	99.999 %	0.0001 %
Neural Network Classifiers	99.107 %	0.893 %

Table 6-3: MIDECCS training set results, per scenario, known-areas only

MIDECC Training Set ¹	True Positives	False Positives
Even Bright	99.383 %	0.470 %
Even Dim	99.532 %	0.337 %
Even Medium	98.608 %	1.386 %
Harsh Moderate 1	90.831 %	1.126 %
Harsh Moderate 2	88.047 %	3.036 %
Harsh Severe 1	83.498 %	2.857 %
Harsh Severe 2	79.197 %	7.323 %
Harsh Slight 1	88.574 %	3.153 %
Harsh Slight 2	84.448 %	3.421 %
Uneven Moderate 1	98.412 %	1.388 %
Uneven Moderate 2	98.192 %	0.882 %
Uneven Severe 1	98.019 %	1.491 %
Uneven Severe 2	88.906 %	2.962 %
Uneven Slight	98.610 %	1.287 %

Table 6-4: Classifier test set results, known-areas only

Test Set	True Positives	False Positives
MIDECC System ¹	92.447 %	2.223 %
Naïve Bayes Classifier	98.555 %	1.448 %
J48 Tree Classifier	99.852 %	0.148 %
Random Tree Classifier	99.713 %	0.288 %
Neural Network Classifier	98.924 %	1.077 %

Table 6-5: MIDECCS test set results, per scenario, known areas only

MIDECC Test Set ¹	True Positives	False Positives
Even Bright	99.375 %	0.547 %
Even Dim	99.508 %	0.329 %
Even Medium	98.562 %	1.430 %
Harsh Moderate 1	89.984 %	1.856 %
Harsh Moderate 2	85.729 %	3.010 %
Harsh Severe 1	83.572 %	3.030 %
Harsh Severe 2	83.423 %	4.424 %
Harsh Slight 1	89.320 %	3.074 %
Harsh Slight 2	83.419 %	4.037 %
Uneven Moderate 1	98.447 %	1.370 %
Uneven Moderate 2	98.342 %	0.807 %
Uneven Severe 1	97.924 %	1.670 %
Uneven Severe 2	90.068 %	1.610 %
Uneven Slight	98.477 %	1.509 %

It is noted that the system struggles to classify colours in scenarios that include the halogen floodlight, designated with ‘Harsh’ in the name. This may be due to the classification process assuming a normal distribution, however this is studied further in Section 6.4.1. Selected scenarios from the table above are discussed over the new few pages, highlighting the strengths and weaknesses of the system.

¹ Note the MIDECC system results do not sum to 100%, discussed at the beginning of this chapter.

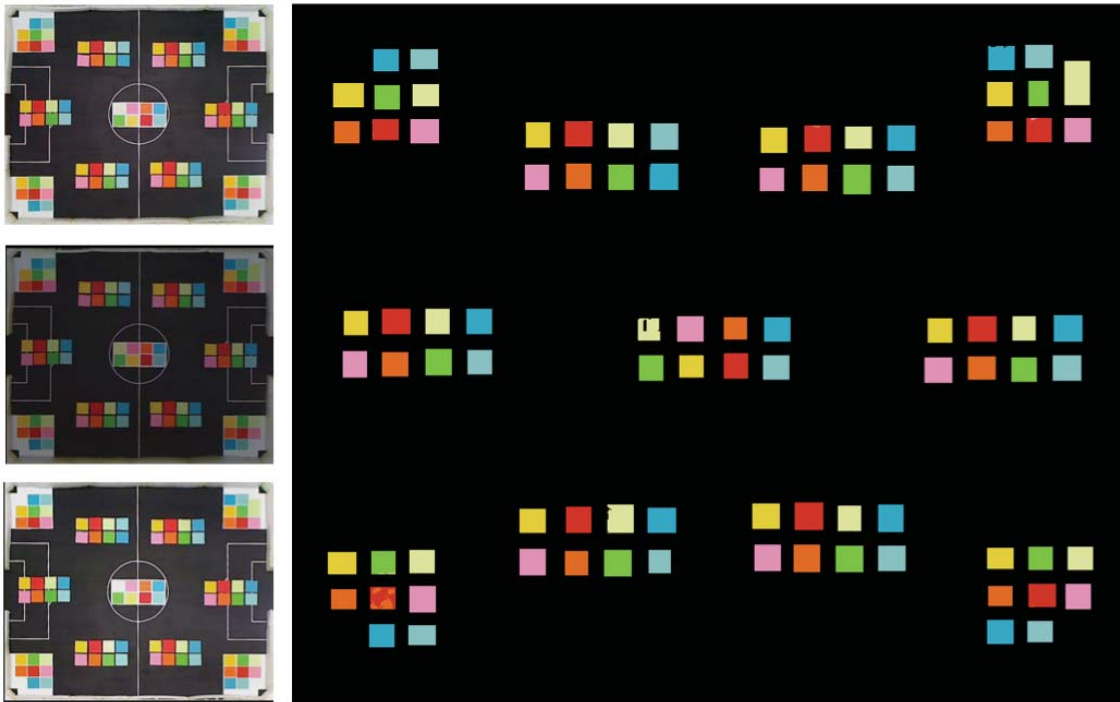


Figure 6-1: Inputs (left) and MIDECCS result (right) of 'Even Bright' training scenario, known-areas only

Above, the output for the relatively easy-to-classify scenario ‘Even Bright’ from the training set results in a high classification value of 99.383% with low false positives of 0.470%.

Table 6-6: MIDECCS Confusion matrix for 'Even Bright' training scenario, known-areas only

MIDECCS Classified As:								Actual Colour Patch:
5794								Orange
	6128							Light Blue
212		6308	7					Red
			6504					Pink
	4			6565				Dark Blue
					6304			Light Green
						6178		Yellow
					14		6339	Dark Green

The confusion matrix indicates the system classified 212 pixels incorrectly as orange, where the correct colour is red. This can be seen in the red colour patch in the lower left corner of the output image.

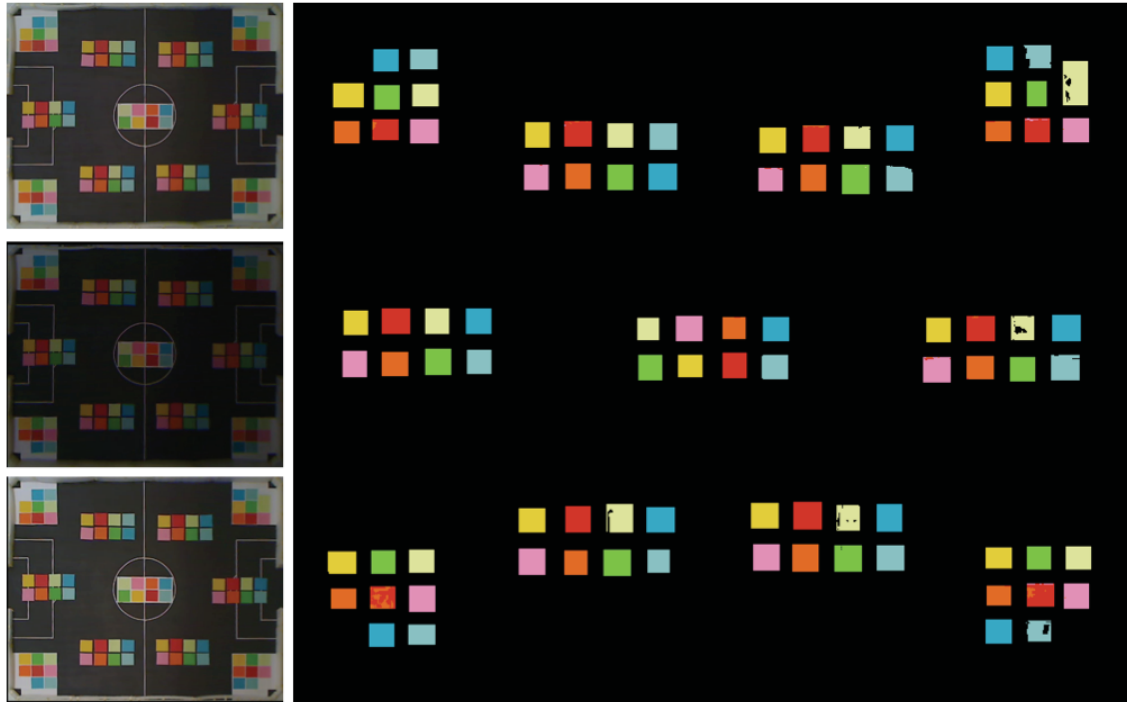


Figure 6-2: Inputs (left) and MIDECCS result (right) of 'Uneven Moderate 2' test scenario, known-areas only

The 'Uneven Moderate 2' test scenario provides a greater challenge for the system, as the right edge of the scene is moderately darker than the left. This can be seen by the input images above, where the colour patches on the right is easily visible from Camera C, while too dark to see in Camera B. This scenario scored 89.342% accuracy, with 0.807% false positives. These results are visible by no large patches of missing or inaccurate colour. It is noticeable that the lighter colours, light green and light blue, begin to start misclassifying at this low light level. Although not classified as another colour, the system simply does not classify as any result.

Table 6-7: MIDECCS Confusion matrix for 'Uneven Moderate 2' test scenario, known-areas only

MIDECCS Classified As:								Actual Colour Patch:
5769		25						Orange
	5911							Light Blue
300		6181	46					Red
9		26	6469					Pink
				6579				Dark Blue
					6160			Light Green
						6176		Yellow
					1		6350	Dark Green



Figure 6-3: Inputs (left) and MIDECCS result (right) of 'Harsh Moderate 1' test scenario, known-areas only

The 'Harsh Moderate 1' test scenario highlights the system struggling under extremely bright conditions. The halogen light in the upper right corner results in most colours in that area becoming completely washed out, however the system continues to classify blue and light blue. Overall, this scenario scores a slightly disappointing total of 89.984% true positives and 1.856% false positives.

Table 6-8: MIDECCS Confusion matrix for 'Harsh Moderate 1' test scenario, known-areas only

MIDECCS Classified As:								Actual Colour Patch:
5338								Orange
	5935							Light Blue
147		5413	421					Red
		6	5976					Pink
	300			6254				Dark Blue
					5037			Light Green
1					46	5569		Yellow
					15		5858	Dark Green

Table 6-9: MIDECCS Confusion matrix for all 14 test scenarios, known-areas only

MIDECCS Classified As:								Actual Colour Patch:
76254		1778						Orange
	79578			132			6	Light Blue
3748		83850	822					Red
1876		2579	82212					Pink
	2950			88588				Dark Blue
					76704	21	730	Light Green
352					558	80880		Yellow
					142		84641	Dark Green

Table 6-10: MIDECCS Confusion matrix for all 14 test scenarios, known-areas only

MIDECCS Classified As:								Actual Colour Patch:
76730		1178	1					Orange
	80467			80			4	Light Blue
4021		82435	1404					Red
1289		1664	83567					Pink
	2834			88826				Dark Blue
					75779	43	955	Light Green
212					584	81332		Yellow
	2				204		84526	Dark Green

6.1.1 Naïve Bayes Classifiers

The Naïve Bayes classifier appears to classify colours correctly, reporting an overall correct percentage of 98.47% for the test set and 98.56% for the training set.

Looking closer at the confusion matrix, it appears the classifier struggles to differentiate between similar colours.

The classifier took 2.34 seconds to train from the 706,034 training data points.

Table 6-11: Naïve Bayes Confusion matrix for all 14 training scenarios, known-areas only

















Naïve Bayes Classified As:								Actual Colour Patch:
								
78524		2372	220					Orange
	85523			153	55		61	Light Blue
2238		88201	827	12				Red
47		15	90994					Pink
	2719			89387				Dark Blue
39			32	5	88759	311		Light Green
362	2				167	85836	125	Yellow
	19				861		88062	Dark Green

Table 6-12: Naive Bays Confusion matrix for all 14 test scenarios, known-areas only

Naïve Bayes Classified As:								Actual Colour Patch:
								
78851		2035	228	2				Orange
	85535			165			60	Light Blue
2213		88400	765					Red
39		18	90999					Pink
	2430			89676	369			Dark Blue
21			9	3	85942	369		Light Green
335					97	85942	118	Yellow
	9				1246		87687	Dark Green

6.1.2 J48 Tree Classifiers

The J48 classifier took 53.75 seconds to train the 531-branch decision tree. Reporting a 99.99% correct classification rate for the training set and slightly lower 99.85% rate for the test set, the J48 classifier correctly distinguished similar colours, with a very small amount of misclassifications in the test set confusion matrix.

Table 6-13: J48 Confusion matrix for all 14 training scenarios, known-areas only

J48 Classified As:								Actual Colour Patch:
81108		8						Orange
	85768			5	17		2	Light Blue
8		91367	3					Red
		1	91051		4			Pink
	22			92084				Dark Blue
	5			16	89140	4	3	Light Green
					16	86476		Yellow
	3				3		88936	Dark Green

Table 6-14: J48 Confusion matrix for all 14 test scenarios, known-areas only

J48 Classified As:								Actual Colour Patch:
80995		86	1		33	1		Orange
	85632			140	4		16	Light Blue
129		91190	59					Red
		2	91048		6			Pink
	131			91974	1			Dark Blue
	20				89039	24	69	Light Green
2			2		296	86192		Yellow
	7			1	13		88921	Dark Green

6.1.3 Random Tree Classifiers

The Random Tree classifier took 14.3 seconds to build a 3,783-branch decision tree. The decision tree reclassified the training set with almost no misclassifications at 99.99% success, however the test set introduced a very small amount of misclassifications at 99.71% success.

Table 6-15: Random Tree Confusion matrix for all 14 training scenarios, known-areas only

















Random Tree Classified As:								Actual Colour Patch:
								
81116								Orange
	85791				1			Light Blue
		91378						Red
			91056					Pink
				92106				Dark Blue
					89152			Light Green
						86492		Yellow
							88942	Dark Green

Table 6-16: Random Tree Confusion matrix for all 14 test scenarios, known-areas only

Random Tree Classified As:								Actual Colour Patch:
								
80775		331			10			Orange
	85385	2	38	226	130		11	Light Blue
81	25	912241	13		18			Red
2	1	339	90499		215			Pink
	218	2	4	91817	65			Dark Blue
3	22				88994	82	51	Light Green
3	6				72	86407	4	Yellow
	12				44		88886	Dark Green

6.1.4 Neural Network Classifiers

The Neural Network training process was run on a simple single-layer neural network, depicted on Page 96. The network was left to train for 20 epochs, totalling 4 minutes and 22 seconds of processing, with a learning rate of 0.3. While this is considered a rudimentary Neural Network, this network was purposely configured to train for a limited number of epochs. While it is possible to configure more layers or train the network for longer lengths of time, the performance obtained by this simple Neural Network provides sufficient comparison data for this thesis.

Despite the short training process, the network still managed to classify colours with considerable accuracy. Evaluating the training set resulted in a 99.11% success rate, while evaluating the test set resulted in a slight decrease to 98.92% success rate.

Table 6-17: Neural Networks Confusion matrix for all 14 training scenarios, known-areas only

















Neural Networks Classified As:								Actual Colour Patch:
								
794974	49	1426		5	5		137	Orange
	85642	8	5	28	11	7	91	Light Blue
136		91213	21	8				Red
	10	19	91027					Pink
	322		633	91151				Dark Blue
1944	33	971	199	18	85986	21	156	Light Green
	92		21		33	68318	28	Yellow
	14			20	5	6	88897	Dark Green

Table 6-18: Neural Networks Confusion matrix for all 14 test scenarios, known-areas only

Neural Networks Classified As:								Actual Colour Patch:
								
79456	48	1209		6	8	28	361	Orange
	84896	3	3	738	50		102	Light Blue
113		91228	33	1				Red
	10	148	90898					Pink
	444		560	91102				Dark Blue
1804	30	978	183	7	85997	16	137	Light Green
	93		19		381	85953	46	Yellow
	8			19	6	5	88904	Dark Green

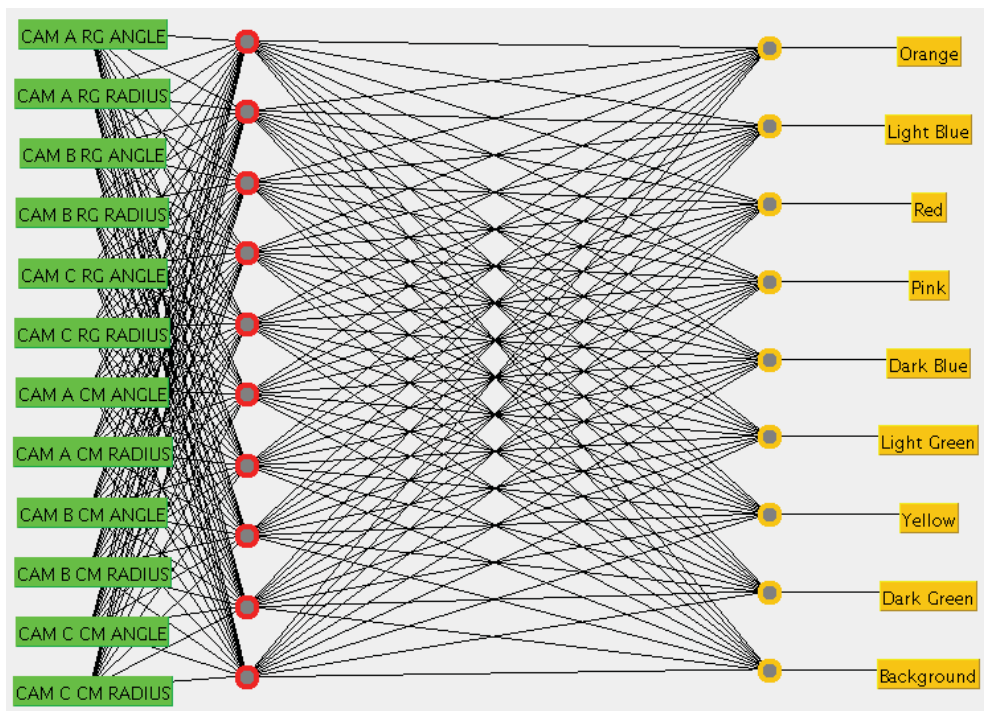


Figure 6-4: Single Layer Neural Network, generated and trained by WEKA

6.2 System Results; Known-Areas and Background

The MIDECC system did not require retraining, as the system may be run on input images with its' current pie slice classifiers. The four WEKA classifiers are to be retrained using the following data set, where the colour patch data is identical to Section 6.1.

Table 6-19: Training sample sizes, including background

Class	Number of Samples
Orange	81,116
Light Blue	85,792
Red	91,378
Pink	91,056
Dark Blue	92,106
Light Green	89,152
Yellow	86,492
Dark Green	88,942
Background	4,000,024
	4,706,058

The number of background instances for training have been reduced from the actual sampled number of 5,740,504 due to computational memory limitations. This was done by removing duplicate values from the data set. Despite the greater amount of background samples, the MIDECC system would not overfit with background classifications as training data which does not match a classifier being trained is discarded.

Table 6-20: Classifier training set results, including background

Training Set	True Positives	False Positives
MIDECC System ¹	92.447 %	4.387 %
Naïve Bayes Classifier	99.344 %	0.656 %
J48 Tree Classifier	99.920 %	0.080 %
Random Tree Classifier	99.899 %	0.101 %
Neural Network Classifier	99.842 %	0.158 %

Table 6-21: MIDECCS training set results, per scenario, including background

MIDECC Training Set ¹	True Positives	False Positives
Even Bright	99.383 %	3.528 %
Even Dim	99.532 %	0.744 %
Even Medium	98.608 %	2.191 %
Harsh Moderate 1	90.831 %	3.262 %
Harsh Moderate 2	88.047 %	6.448 %
Harsh Severe 1	83.498 %	5.046 %
Harsh Severe 2	79.197 %	10.511 %
Harsh Slight 1	88.574 %	6.554 %
Harsh Slight 2	84.448 %	11.035 %
Uneven Moderate 1	98.412 %	2.328 %
Uneven Moderate 2	98.192 %	2.356 %
Uneven Severe 1	98.019 %	1.925 %
Uneven Severe 2	88.906 %	3.145 %
Uneven Slight	98.610 %	2.346 %

Due to computational memory limitations, the test set was split across two data files and results combined. The data sets were ordered by colour, followed by background samples. The data was then split down the middle, creating two smaller, manageable files to load into WEKA. The first data file consisted of all colours and some background samples, while the second data file consisted of background samples only. As the WEKA classifiers do not change during testing, combined results obtained with two files are directly comparable to a system which tests with one data file.

The testing of the WEKA classifiers incorporated all 5,740,504 background colour samples to produce the following results.

Table 6-22: Classifier test set results, including background

Test Set	True Positives	False Positives
MIDECC System ¹	92.582 %	4.247 %
Naïve Bayes Classifiers	99.266 %	0.834 %
J48 Tree Classifiers	99.910 %	0.090 %
Random Tree Classifiers	99.901 %	0.099 %
Neural Network Classifiers	99.754 %	0.246 %

Table 6-23: MIDECCS test set results, per scenario, including background

MIDECC Test Set ¹	True Positives	False Positives
Even Bright	99.375 %	3.658 %
Even Dim	99.508 %	0.724 %
Even Medium	98.562 %	2.467 %
Harsh Moderate 1	89.984 %	4.065 %
Harsh Moderate 2	85.729 %	6.284 %
Harsh Severe 1	83.572 %	5.177 %
Harsh Severe 2	83.423 %	7.727 %
Harsh Slight 1	89.320 %	7.010 %
Harsh Slight 2	83.419 %	11.263 %
Uneven Moderate 1	98.447 %	2.044 %
Uneven Moderate 2	98.342 %	2.338 %
Uneven Severe 1	97.924 %	1.937 %
Uneven Severe 2	90.068 %	1.921 %
Uneven Slight	98.477 %	2.847 %

Again, the ‘Harsh’ scenarios reduce the overall classification scoring for the MIDECC system. While staying at a similar percentage of classification scoring, it is noticeable that the WEKA classifiers have a greater amount of false positives, albeit by a small amount. The reasoning behind the MIDECC system lower true positives is to be discussed in Section 6.4.1.

The same scenarios highlighted in the previous section will be explored again with the addition of background pixels to compare results. The confusion matrices for the

¹ Note the MIDECC system results do not sum to 100%, discussed at the beginning of this chapter.

MIDECC system are the size of 9 rows by 9 columns, however much of the 9th column will remain empty. While it is possible the system mistakenly classifies the background as a colour, it does not classify a background class.

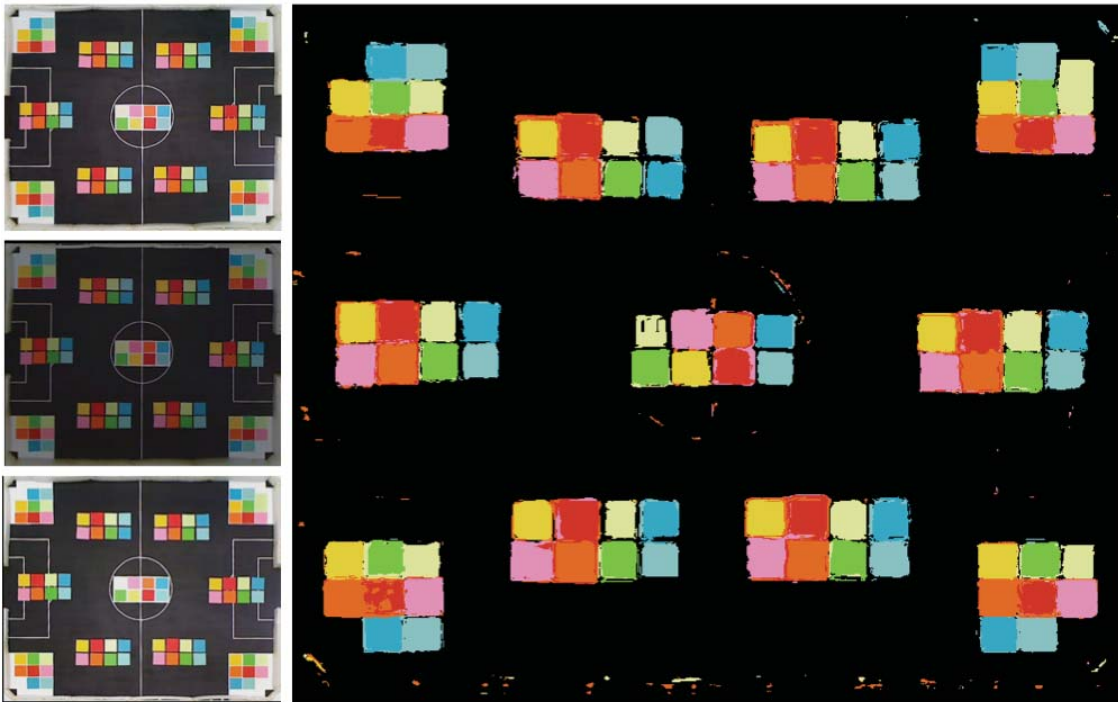


Figure 6-5: Inputs (left) and MIDECCS result (right) of 'Even Bright' training scenario, including background

Introduction of the background pixels has resulted in the system increasing the false positives from 0.470% to 3.528% for the 'Even Bright' training scenario. This increase in false positives is visible with the lighter colours such as yellow, orange and red being classified in the background. A total of 1,542 pixels were incorrectly classified in the background.

Table 6-24: MIDECCS Confusion matrix for 'Even Bright' training scenario, including background

MIDECCS Classified As:									Actual Colour:
								BG	
5794									Orange
	6128								Light Blue
212		6308	7						Red
			6504						Pink
	4			6565					Dark Blue
					6304				Light Green
						6178			Yellow
					14		6339		Dark Green
2822	95	1495	882	933	919	337	171	408494	Background

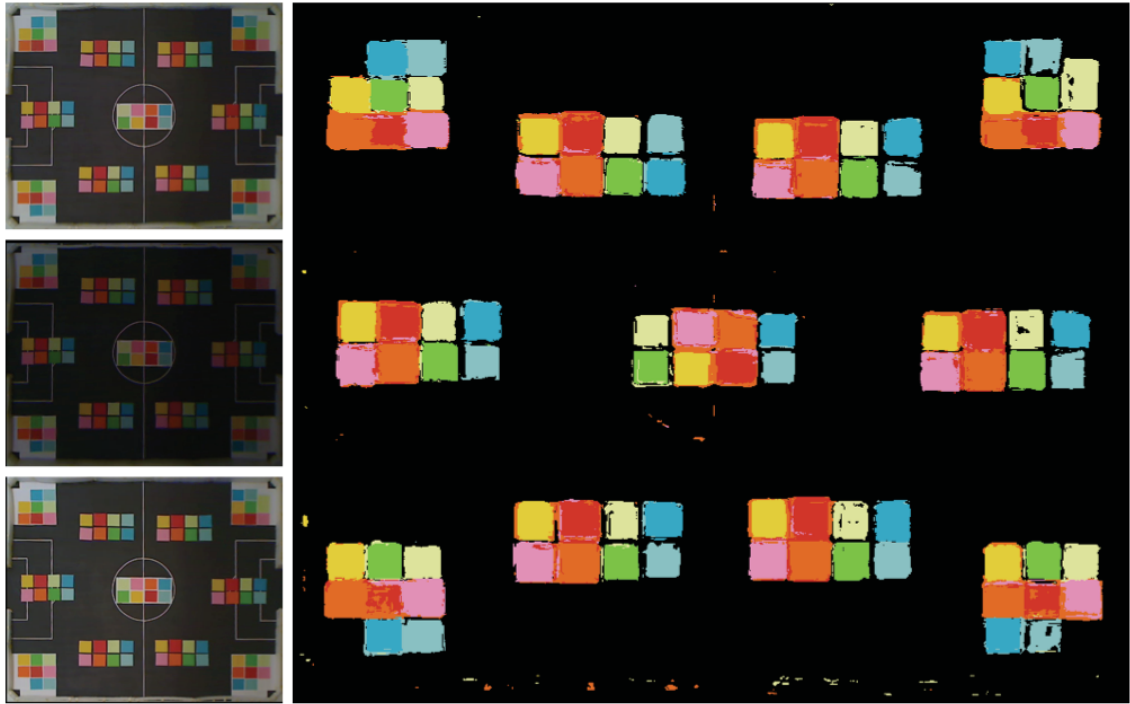


Figure 6-6: Inputs (left) and MIDECCS result (right) of 'Uneven Moderate 2' test scenario, including background

The 'Uneven Moderate 2' test scenario classifies the same true positive and false positive results of 89.342% and 2.338%, respectively. Adding the background samples resulted in 772 background pixels being incorrectly classified as a colour.

Table 6-25: MIDECCS Confusion matrix for Uneven Moderate 2' test scenario, including background

MIDECCS Classified As:									Actual Colour:
								BG	
5769		25							Orange
	5911								Light Blue
300		6181	46						Red
9		26	6469						Pink
				6579					Dark Blue
					6160				Light Green
						6176			Yellow
					1		6350		Dark Green
408		54	25		198	87		409264	Background

Once again the highest classified colour for the background is orange, followed by light green.

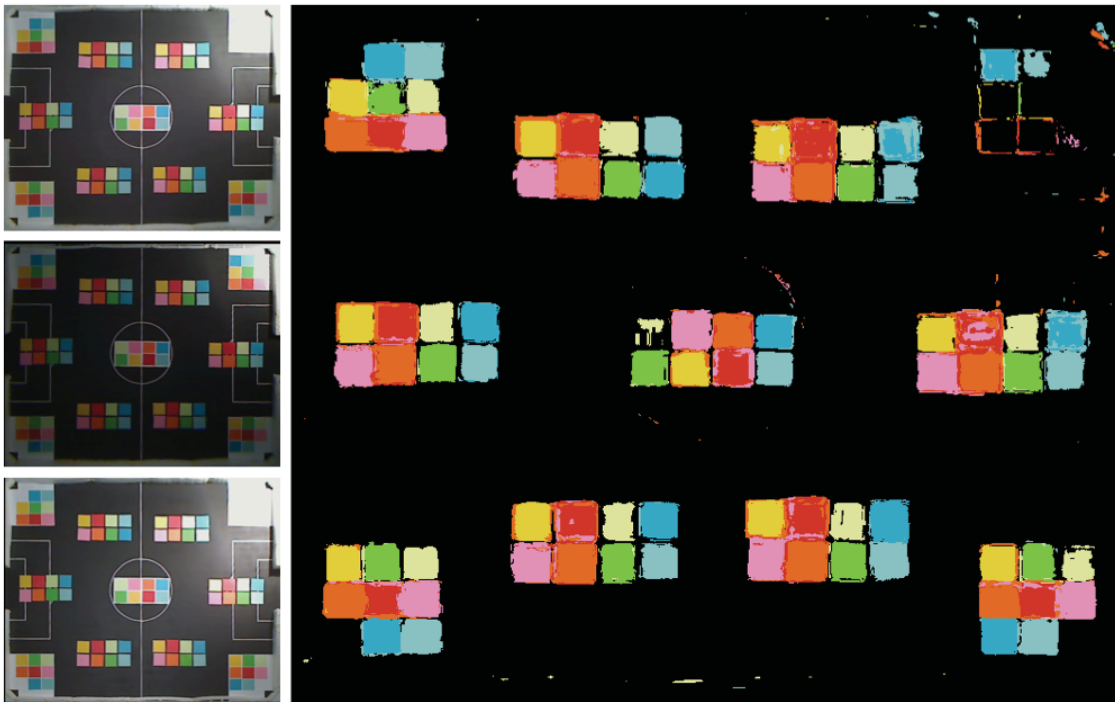


Figure 6-7: Inputs (left) and MIDECCS result (right) of 'Harsh Moderate 1' test scenario, including background

Under the halogen flood lamp in the 'Harsh Moderate 1' scenario, the background is misclassified 1,114 times. This is especially visible around the edges of the brightly lit corner patches, where the light and slight misalignment of cameras have resulted in a very thin strip of pixels being classified.

Table 6-26: MIDECCS Confusion matrix for 'Harsh Moderate 1' test scenario, including background

MIDECCS Classified As:									Actual Colour:
								BG	
5338									Orange
	5935								Light Blue
147		5413	421						Red
		6	5976						Pink
	300			6254					Dark Blue
					5037				Light Green
1					46	5569			Yellow
					15		5858		Dark Green
337	304	125	120	28	169	5	26	408922	Background

Lighter colours such as orange, pink, light blue and light green have been classified more often as the background compared to other colours.

Table 6-27: MIDECCS Confusion matrix for all 14 training scenarios, including background

MIDECCS Classified As:									Actual Colour:
								BG	
76254		1778							Orange
	79578			132			6		Light Blue
3748		83850	822						Red
1876		2579	82212						Pink
	2950			88588					Dark Blue
					76704	21	730		Light Green
352					558	80880			Yellow
					142		84641		Dark Green
7767	1622	1252	991	195	2523	726	204	5725224	Background

Table 6-28: MIDECCS Confusion matrix for all 14 test scenarios, including background

MIDECCS Classified As:									Actual Colour:
								BG	
76730		1178	1						Orange
	80467			80			4		Light Blue
4021		82435	1404						Red
1289		1664	83567						Pink
	2834			88826					Dark Blue
					75779	43	955		Light Green
212					584	81332			Yellow
	2				204		84526		Dark Green
7888	1702	1272	1040	203	2180	1052	176	5724991	Background

6.2.1 Naïve Bayes Classifiers

Taking 18.54 seconds to train, the Naïve Bayes classifier reports a greater true positive percentage when trained with the background pixels compared to known-areas only. This may be due to more ‘splits’ in the colour space data, allowing greater definition of each feature due to the Naïve assumptions the classifier makes during training. The decrease of false positives to 0.656% also supports this theory, where it is likely that the Naïve Bayes classifier may now make better assumptions due to the greater spread of training data.

Table 6-29: Naive Bays Confusion matrix for all 14 training scenarios, including background

Naïve Bayes Classified As:									Actual Colour:
								BG	
78518		2372	222					4	Orange
	81309			11	7		14	4451	Light Blue
2334		88201	817	6				20	Red
47		15	87936					3058	Pink
	1681			89267				1158	Dark Blue
					84765	9	6	4372	Light Green
262					93	85225	6	906	Yellow
					215		87615	1112	Dark Green
2822	95	1495	882	933	919	337	171	3992370	Background

Table 6-30: Naive Bays Confusion matrix for all 14 test scenarios, including background

Naïve Bayes Classified As:									Actual Colour:
								BG	
78833		2035	223					15	Orange
	81442			23	4		14	4309	Light Blue
2211		88402	764					1	Red
39		19	87806					3193	Pink
	1427			89588				1091	Dark Blue
					84761	33	1	4357	Light Green
245					55	85342	2	848	Yellow
	2				327		86432	981	Dark Green
2902	85	1545	916	1090	993	373	139	5732461	Background

6.2.2 J48 Tree Classifier

The J48 classifier generated a 587-leaf decision tree in just under 12 minutes. This long processing time resulted in similar scores in true positive and false positives compared to the no-background data set. This highlights the optimisations present in the processing of this classifier, being able to output a high accuracy score with a variety of information such as the 4.7 million background data samples.

Table 6-31: J48 Confusion matrix for all 14 training scenarios, including background

J48 Classified As:									Actual Colour:
								BG	
81103		13							Orange
	85687			8	20		2	75	Light Blue
9		91367	2						Red
1			91044		6	1		4	Pink
	21			92066				19	Dark Blue
	3		2		89090	3		54	Light Green
			1		13	86477		1	Yellow
	6				1		88934	1	Dark Green
	10	1	2	2	14	1	1	3999993	Background

Table 6-32: J48 Confusion matrix for all 14 test scenarios, including background

J48 Classified As:									Actual Colour:
								BG	
80876		97	1		138			4	Orange
	85285			52	105	1	10	339	Light Blue
151		90953	274						Red
1		6	90244		76	1		728	Pink
	93			91925	6			82	Dark Blue
	29		8		88872	93	19	131	Light Green
					45	86437		10	Yellow
	228			3	2		88704	5	Dark Green
4	76	16	20	43	171	16	2	5740156	Background

6.2.3 Random Tree Classifier

Building a large 7,199 branch decision tree, the Random Tree classifier took 1 minute and 11 seconds to train with background samples. Assessing the Training Set, the classifier scored extremely highly, expected for a large decision tree. Re-evaluation on the test set resulted in minimal incorrect background classifications, however due to the scattered data samples in the confusion matrix it can be concluded that the Random Tree classifier has trouble distinguishing between the lighter colours.

Table 6-33: Random Tree Confusion matrix for all 14 training scenarios, including background

Random Tree Classified As:									Actual Colour:
								BG	
81116									Orange
	85790			1	1				Light Blue
		91378							Red
			91053		3				Pink
				92106					Dark Blue
					89152				Light Green
						86492			Yellow
							88942		Dark Green
								4000024	Background

Table 6-34: Random Tree Confusion matrix for all 14 test scenarios, including background

Random Tree Classified As:									Actual Colour:
								BG	
80746		258			50	41		21	Orange
	85122			375	50		19	226	Light Blue
149	2	91192	19		3			13	Red
8	5	116	90644		261			22	Pink
	135	194	40	91426	12	28	17	254	Dark Blue
	99				88940	33	20	60	Light Green
27	2			1	409	86001	37	15	Yellow
	20			1	43	26	88848	2	Dark Green
23	341	21	67	101	285	27	13	5739626	Background

6.2.4 Neural Network Classifier

Training the single layer network for 20 epochs at the learning rate of 0.3, the Neural Network classifier took just under 12 minutes with the background samples. Evaluating the training set resulted in 99.84% correct classifications with minimal (0.158%) false positives. Re-evaluating the test set with the total number of background samples produces 99.75% true positives and 0.246% false positives.

Table 6-35: Neural Network Confusion matrix for all 14 training scenarios, including background

Neural Network Classified As:									Actual Colour:
								BG	
79811	183	193	38				659	232	Orange
	85509			7	30		95	151	Light Blue
367		90965	39					6	Red
7			91026					23	Pink
	966			91033			25	82	Dark Blue
	4				85964	7	1877	1300	Light Green
					211	85917	2	362	Yellow
	6				182	48	88673	33	Dark Green
13	5	14	101	9	114	44	1	3999723	Background

Table 6-36: Neural Network Confusion matrix for all 14 test scenarios, including background

Neural Network Classified As:									Actual Colour:
								BG	
79766	178	163	116				717	176	Orange
	85267			161	34		109	221	Light Blue
417	1	90612	345					3	Red
124		140	90755					37	Pink
	1102			90805			6	193	Dark Blue
					85700	9	1936	1507	Light Green
					262	85864	2	364	Yellow
	5				209	68	88634	26	Dark Green
24	3	19	125	17	91	49	3	5740173	Background

6.3 Summary

After analysing the confusion matrices for each classifier, it is noticeable that if a pixel is deemed too bright or too dark, the MIDECC system will not classify any colour. This is visible by the confusion matrix data being grouped together, compared to being spread over many different colours. The grouping of data on the confusion matrix indicates the MIDECC system classifying similar colours distinctly, minimising misclassifications.

The Naïve Bayes classifier generated a strong true positive classification value, however of all WEKA classifiers, it generated the most false positives. The false positive rate did drop significantly once the background data was added, indicating that the assumption-based training works better with the additional background data for this task.

The J48 Tree classifier proved to be robust with both the training and test sets, minimising classification errors. The J48 classifiers were let down by the longer training processing time, nevertheless the classifiers do perform well, with greater than 99% success.

Training very quickly, the Random Tree classifiers built a very large decision tree which would be expected to return its' 99.99% success rate for the training set. This value falls only slightly with a small number of misclassifications for the testing set. Light blue and light green are two colours which have a multitude of misclassified colours, compared to the robust detection of the other colours.

Neural Networks classifiers trained 20 epochs, taking four and a half minutes for no background data and just under 12 minutes with background data. The single layer network resulted in a 98.9% success rate, although similar colours such as red and pink did cause the majority of misclassifications. It is noted that the Neural Network classifier was excellent at separating the background compared to the coloured patches, correctly classifying nearly all of the background pixels as the background class.

6.4 Discussion

The MIDECC system has been tested with 14 different testing scenarios of slight to severe illumination difference. The overall system confusion matrix indicates minimal misclassifications of similar colours, with orange and red being the most troublesome. Yellow and dark blue, followed by green are the best classified colours. Dim scenarios such as ‘Even Dim’ and ‘Uneven Severe’ are classified excellently, excelling over a single camera setup.

The overall classification result of over 92% for both training and test data sets highlight the effectiveness of a relatively simple classification system coupled with the probabilistic approach in resolving multiple classifier matches.

It is also notable that the MIDECC system is less likely to misclassify vastly different colours, for example blue and red. This is due to the classifiers’ ordering of data in the hyperspace, placing similar colours next to one another. The fundamental layout of the red/green and cyan/magenta colour space result in the system being likely to classify the correct colour, compared with an arbitrary classification space, used by the WEKA classifiers. The MIDECC system did struggle under harsh illumination, where two out of the three cameras could not classify a colour. Often the lighter colours such as light blue and light green are more likely to be misclassified, as these colours are located near the centre of the colour spaces. This location results in the possibility of being disregarded by the background filtering process or being less saturated, outside the pie slice classifier.

The following sub-sections discuss the limitations and possible Fuzzy Colour Contrast Fusion (FCCF) [1] implementation results.

6.4.1 Harsh Lighting Limitations

After further testing, the classifier construction method appears to be the culprit for the misclassifications in the harsh lighting scenarios. The system uses the assumption of a normally distributed spread of data when generating the classifier.

Take the below spread of data that may be generated during the training process for Red:

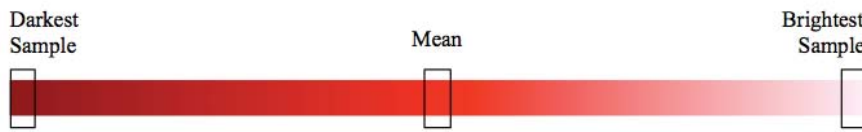


Figure 6-8: Example of sample data for Red

The classifier construction process states that the system is to use a single standard deviation either side of the mean, covering an estimated 68% of sample data, shaded green, below.

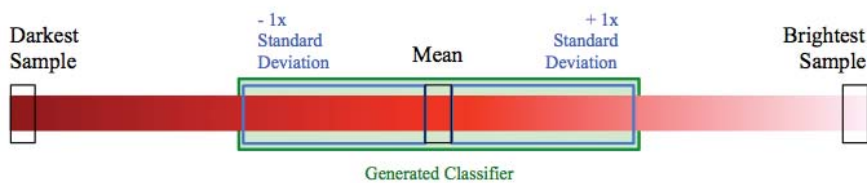


Figure 6-9: Classifier generated for 1x Standard Deviation either side of the mean

During the classification process, the system searches inside the above green 'generated classifier' region, before expanding the search an additional standard deviation in an attempt to match a colour if none is found:

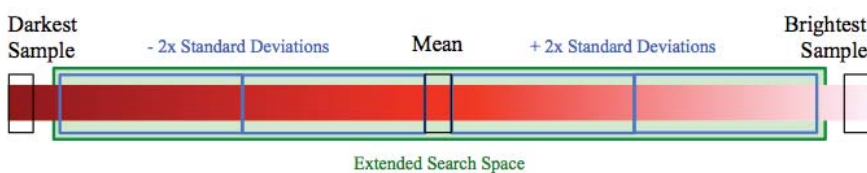


Figure 6-10: Extended search space, generated by 2x Standard Deviations either side of the mean

Due to the calculated standard deviation, this estimated search space will cover approximately 95% of the sample data.

The decision not to extend the search process to a third standard deviation was chosen due to the addition of noise and false positives.

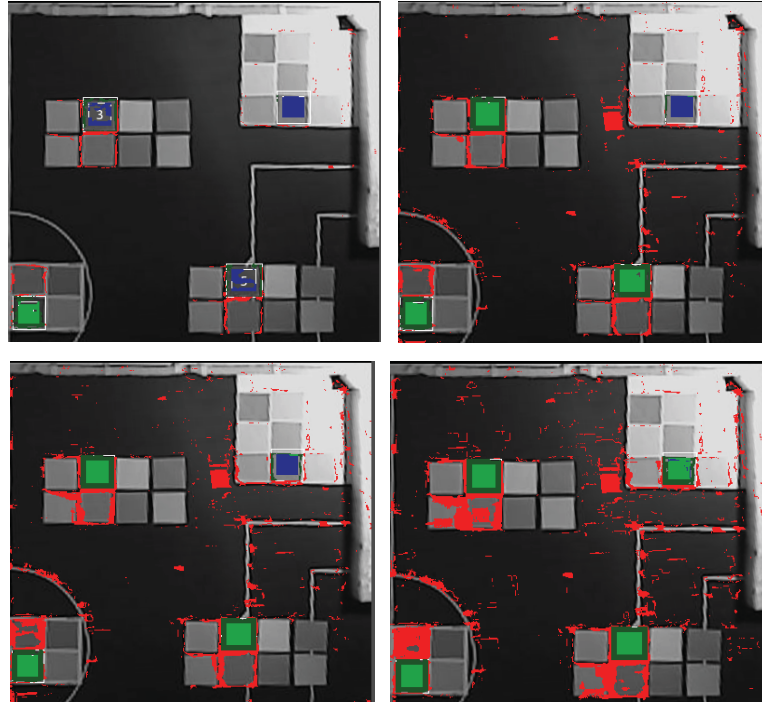


Figure 6-11: Clockwise, from upper left: 1x Standard Deviation, 2x Standard Deviations, 3x Standard Deviations, 4x Standard Deviations

Colour Key:

Correctly Classified	■ or ■
Incorrectly Classified	■
Incorrectly Classified, within angle ranges but outside radius ranges	■

The above samples are statistical coloured output of Camera B, searching for Red in the ‘Harsh Severe’ training scenario.

It is important to ignore the red ‘noise’ in the background of the images, as this is removed by the background filtering further in the pipeline. What can be seen clearly is the misclassifications (shown in red) increasing as the standard deviations increase.

At four standard deviations, the brightest area of the image correctly classifies. This comes at the expense of also classifying most of the orange areas as red also, negating any gains.

Without pre-processing or further experiments into selective standard deviation additions, the brightest illuminated areas in the harsh scenarios are too difficult for the MIDECC system to process reliably.

6.4.2 Classification with FCCF

Fuzzy Colour Contrast Fusion (FCCF), introduced by Reyes and Dadios appeared to be a promising addition to the system, adjusting the contrast for the red, green and blue channels of each camera. The use of fuzzy logic ensures the correct contrast operators are assigned to each camera at the optimum strengths. FCCF was successfully implemented with the MIDECC system, automatically choosing the optimum contrast operators for each colour under each camera by testing each combination.

Many of the colours did not require FCCF correction, with the results of the brute-force testing method returning ‘No Change’ for all three colour channels.

Table 6-37: FCCF Rules derived using brute force, as outlined in [1]

	Camera A (Best Overall)			Camera B (Darker)			Camera C (Brighter)		
	R	G	B	R	G	B	R	G	B
	D1	NC	E1				NC	NC	D1
				D2	NC	NC	D2	E1	E1
				D1	D1	D1			
				NC	NC	E1			
				E1	NC	E2			

Where Dx and Ex denotes contrast degradation or enhancement at the numbered strength, respectively, while NC denotes ‘No Change’. Blank cells were returned as ‘No Change’ and are excluded from the table for ease of visualisation.

The formulas used to calculate contrast degradation and enhancement can be found in Section 3.2.2.

Unfortunately the FCCF process outlined by Reyes and Dadios did not improve results with the proposed system - in fact reducing the total number of classifications. The promising algorithm works excellently for a single camera setup, however further study will be needed for optimum implementation with multiple cameras.

It is thought that this result is due to the incorrect prioritising of cameras. Without FCCF, the normal densities calculation of the angle and radius create a classifier ‘confidence’ level successfully, prioritising the correct colour or camera over another. With FCCF however, colours which have FCCF rules will naturally be a ‘better fit’ to the classifier, generating a greater confidence level. The results of the proposed system including FCCF processing are 89.27% for the training data set and 89.57% for the test data set.

Further study into this issue may include a more selective approach to FCCF, or at the very least weighting classifier priority if the FCCF process has changed the colour significantly.

Chapter 7

7. Conclusions

The Multiple Image Dynamic Exposure Colour Classification (MIDECC) System was developed in an effort to provide reliable colour classification for a scene with extreme variations in lighting conditions.

Development of different sub-systems which pre-process multiple camera images (Section 4.3), remove background samples (Section 4.6) and score unique colour space models (Section 4.5.2) result in a robust system which requires minimal user training parameters. The MIDECC System is quick to train, with the majority of time spent averaging samples or waiting for external camera inputs to stabilise.

Drawing on previous research into a unique classifier [19] and normalised red/green chromaticity [1], a novel complementary cyan/magenta chromaticity is proposed which further assists the system by providing a greater separation between individual classifiers, grouping certain colours together while separating others.

The MIDECC System does not require retraining for a data set that includes background samples, as the background removal process largely deals with the eradication of false positives, while keeping the true positive rate the same as without background input.

When different input images match dissimilar classifiers, a simple normal density function is assessed in the preferred classifiers' colour space. This provides a relatively quick and reliable way of prioritising matches, resulting in a single classifier output.

The results of experiments herald a 92.46% true positive rate, while minimising false positives to 4.25%. The system has been observed to reliably classify colours in illumination ranging from 210 lux to 1,097 lux in the same frame. The lower illumination limit of 210 lux is equivalent to two standard fluorescent tubes at a height of 3 meters illuminating a room totally occluded from sunlight, while the upper limit of 1,097 lux is comparable 14 standard fluorescent tubes and 4 halogen spotlights illuminating the room at the same height.

7.1 Suggestions for Future Work

There are four areas that have been identified for further work:

1. Implementation of a weighted FCCF system, allowing the MIDECC normal density function to correctly prioritise FCCF adjusted colours. Currently, the system incorrectly gives priority to any colour adjusted by FCCF as the normal density function returns a high value. It is expected that this would improve the overall true positive classification rate of the MIDECC System, expanding the illumination classification range.
2. Implementation of a ‘worst case’ classifier matching process, where the pie slices of likely classifiers are expanded to match a coloured pixel.
This would enable the matching of the outliers in extremely bright or dim illuminated regions, working together with the background removal system to reduce additional false positive matches.
3. Conversion of final values into a lookup table, resulting in faster classification times.
Although the lookup table would be large, the MIDECC system could be developed to run real time without complex calculations. At this stage, the experimental implementation of the MIDECC system takes approximately 5 seconds per frame as statistics are recalculated often.
4. Manual reference inputs such as the colour selection may be automated.
Provided known-areas are marked out for the system, all matches for a particular area may be plotted to a colour space, before selecting a group or cluster for the colour exposure selection reference.

Bibliography

- [1] N. H. Reyes and E. P. Dadios, “Dynamic Color Object Recognition Using Fuzzy Logic,” 2003.
- [2] J. Blanc-Talon, A. Kasinski, W. Philips, D. Popescu, and P. Scheunders, “Acquisition of Agronomic Images with Sufficient Quality by Automatic Exposure Time Control and Histogram Matching,” in *Advanced Concepts for Intelligent Vision Systems*, 2013, vol. 8192, pp. 37–48.
- [3] E. H. A and R. N, “OCR Accuracy Improvement on Document Images Through a Novel Pre-Processing Approach,” *Signal Image Process. An Int. J.*, vol. 6, no. 4, pp. 01–18, Aug. 2015.
- [4] A. O. Ercan, A. El Gamal, and L. J. Guibas, “Object Tracking in the Presence of Occlusions via a Camera Network,” in *2007 6th International Symposium on Information Processing in Sensor Networks*, 2007, pp. 509–518.
- [5] G. Garibotto and C. Cibeï, “3D scene analysis by real-time stereovision,” in *IEEE International Conference on Image Processing 2005*, 2005, vol. 2, pp. II–105.
- [6] J. Kronander, S. Gustavson, G. Bonnet, A. Ynnerman, and J. Unger, “A unified framework for multi-sensor HDR video reconstruction,” *Signal Process. Image Commun.*, vol. 29, no. 2, pp. 203–215, Feb. 2014.
- [7] H. E. Smithson, “Sensory, Computational and Cognitive Components of Human Colour Constancy,” *Philos. Trans. Biol. Sci.*, vol. 360, no. 1458, pp. 1329–1346, 2005.
- [8] S. McHugh, “Cambridge in Colour: Digital Camera Sensors,” 2016. [Online]. Available: <http://www.cambridgeincolour.com/tutorials/camera-sensors.htm>. [Accessed: 20-Nov-2015].
- [9] A. L. C. Barczak, T. Susnjak, N. H. Reyes, and M. J. Johnson, “Colour segmentation for multiple low dynamic range images using boosted cascaded classifiers,” in *International Conference Image and Vision Computing New Zealand*, 2013, pp. 136–141.
- [10] “OpenCV Documentation: findHomography(),” 2014. [Online]. Available: http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html. [Accessed: 15-Oct-2015].
- [11] “OpenCV Documentation: warpPerspective(),” 2014. [Online]. Available: http://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html. [Accessed: 15-Oct-2015].
- [12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA Data Mining Software: An Update,” *ACM SIGKDD Explor.*, vol. 11, no. 1, pp.

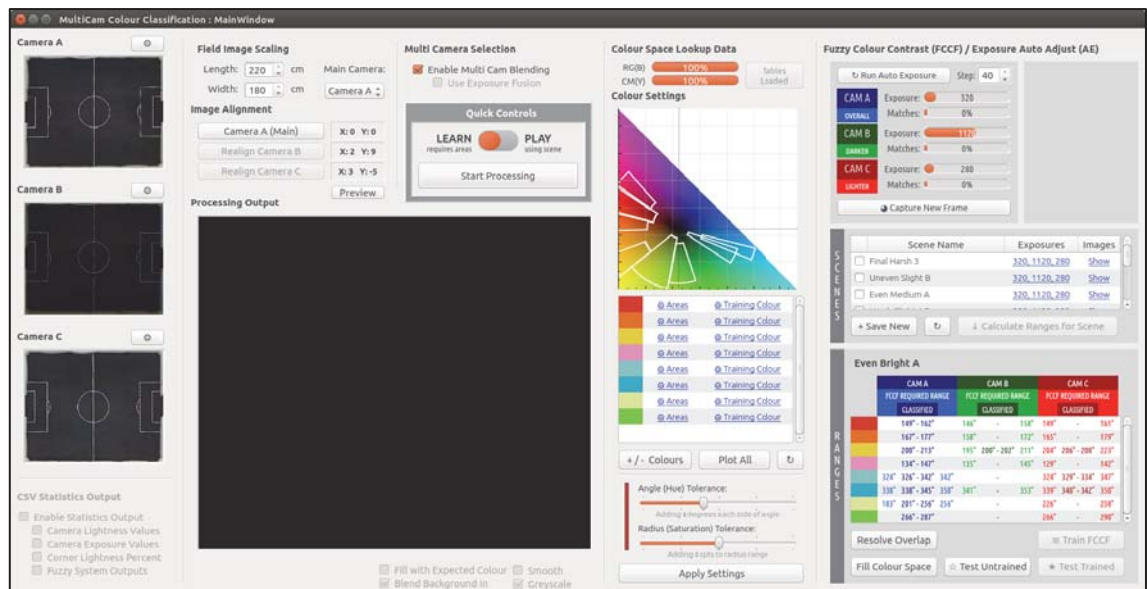
10–18, 2009.

- [13] H. Zhang, “The Optimality of Naive Bayes,” *Proc. Seventeenth Int. Florida Artif. Intell. Res. Soc. Conf. FLAIRS 2004*, vol. 1, no. 2, pp. 1 – 6, 2004.
- [14] E. Keogh, “Naïve Bayes Classifier,” in *Computer Engineering Course, University of California - Riverside*, 2011.
- [15] J. R. Quinlan, *C4.5: Programs for Machine Learning*, vol. 1, no. 3. 1992.
- [16] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, *Top 10 algorithms in data mining*, vol. 14, no. 1. 2008.
- [17] B. Reed, “The height of a random binary search tree,” *J. ACM*, vol. 50, no. 3, pp. 306–332, May 2003.
- [18] D. P. Playne, V. D. Mehta, N. H. Reyes, and A. L. C. Barczak, “Hybrid Fuzzy Colour Processing and Learning,” pp. 386–395, 2008.
- [19] P. J. Thomas, R. J. Stonier, and P. J. Wolfs, “Robustness of colour detection for robot soccer,” in *7th International Conference on Control, Automation, Robotics and Vision, 2002. ICARCV 2002.*, 2002, vol. 3, pp. 1245–1249.
- [20] L. Hildebrand and M. Fathi, “Knowledge-Based Fuzzy Color Processing,” *IEEE Trans. Syst. Man Cybern. Part C (Applications Rev.)*, vol. 34, no. 4, pp. 499–505, Nov. 2004.
- [21] K. Heidary and H. J. Caulfield, “Color classification using margin-setting with ellipsoids,” *Signal, Image Video Process.*, pp. 1–18, 2012.
- [22] N. Kwok, H. Shi, G. Fang, Q. Ha, Y.-H. Yu, T. Wu, H. Li, and T. Nguyen, “Color image enhancement using correlated intensity and saturation adjustments,” *J. Mod. Opt.*, vol. 62, no. 13, pp. 1037–1047, Jun. 2015.
- [23] J. B. Zimmerman, S. M. Pizer, E. V. Staab, J. R. Perry, W. McCartney, and B. C. Brenton, “Evaluation of the effectiveness of adaptive histogram equalization for contrast enhancement,” *IEEE Trans. Med. Imaging*, vol. 7, no. 4, pp. 304–312, 1988.
- [24] “OpenCV Documentation: Image Thresholding,” 2015. [Online]. Available: http://docs.opencv.org/3.1.0/d7/d4d/tutorial_py_thresholding.html. [Accessed: 11-Aug-2015].

Appendix A: Experiment Implementation

The MIDECC System was implemented in C++ in the Qt IDE, allowing rich GUI functionality. Additional libraries were also used, listed below.

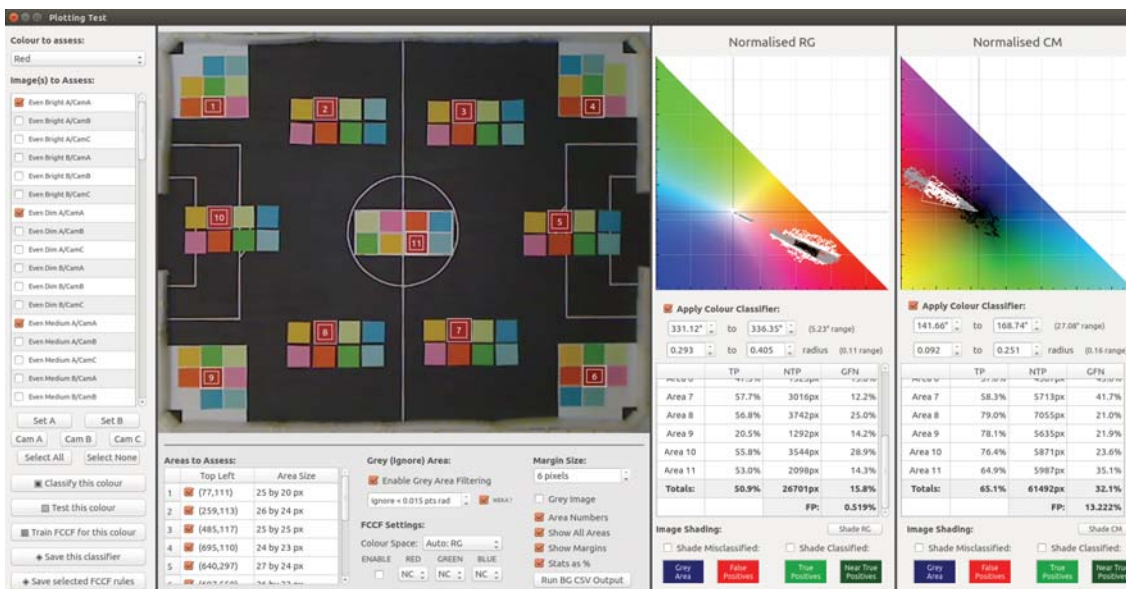
Software	Version	Use in MIDECC System
Qt IDE www.qt.io	5.2.1 GCC 4.8.2 64-bit	<ul style="list-style-type: none"> • C++ Framework • User Interface & Window Management • File I/O • Multithreading • Painting output graphics
OpenCV www.opencv.org	3.0	<ul style="list-style-type: none"> • Camera Image Input • Homography Processing • Image Pre-processing
QCustomPlot www.qcustomplot.com	1.3.1	<ul style="list-style-type: none"> • Colour space graphics & plot display
Video4Linux linuxtv.org	2.0	<ul style="list-style-type: none"> • Camera command



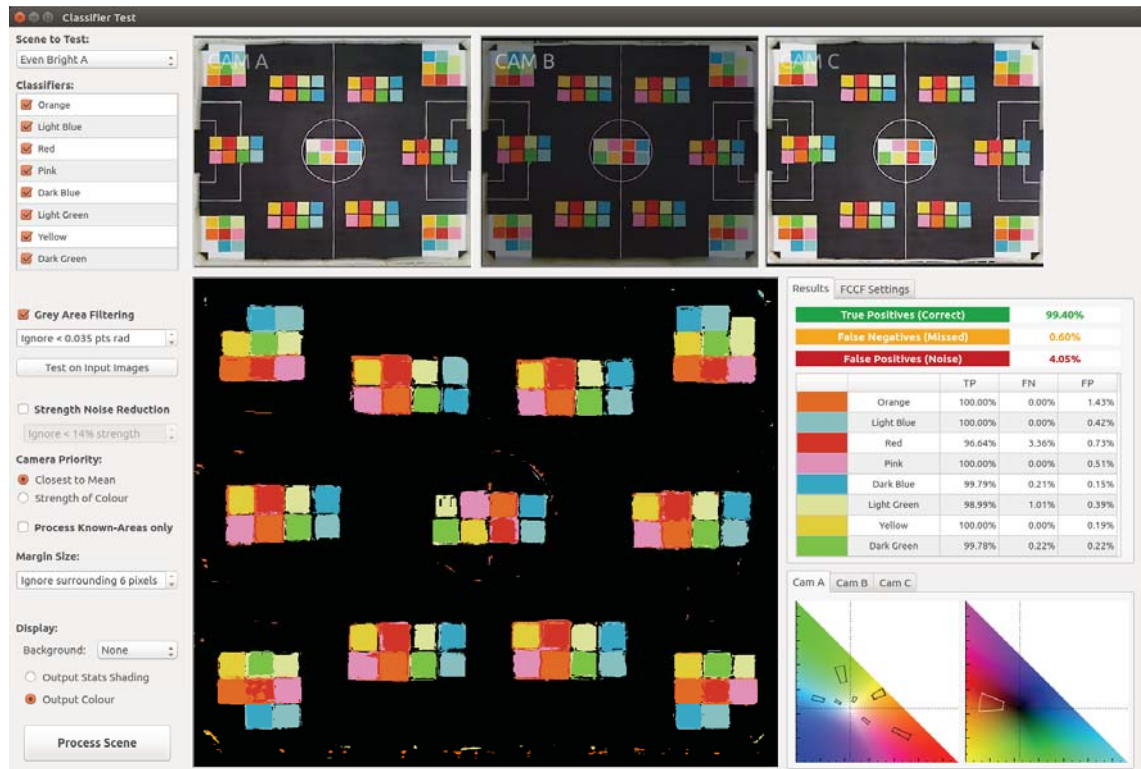
This program facilitates the image pre-processing by manually selected homographic points, scales image based on the real-life measurements entered, then automatically aligns each camera. The exposure selection process is run using the defined colours and areas, before allowing lighting scenarios to be saved to file.



Camera C, shown above, ready for alignment. The homographic points, generated by clicking on the live image, are shown as a red dotted rectangle.



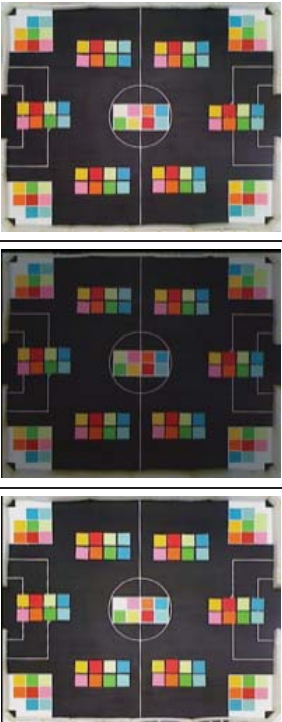
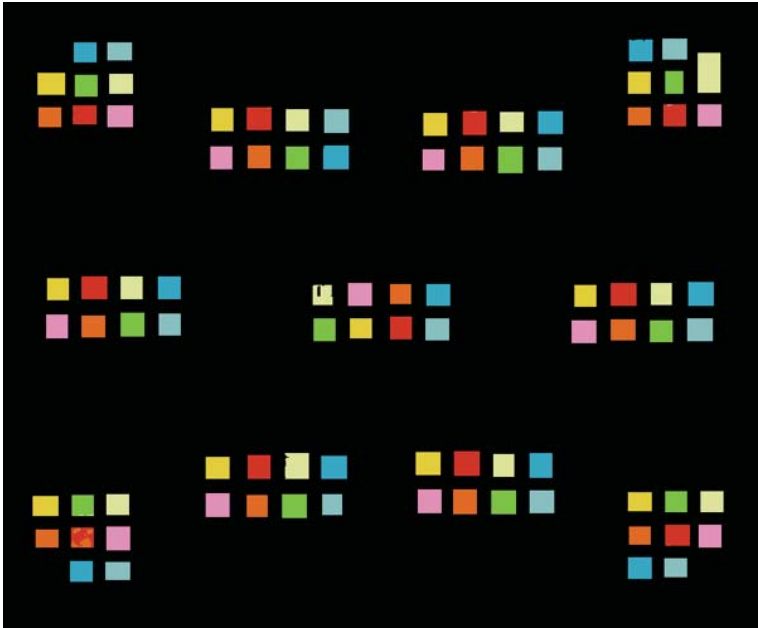
The next program loads the PNG image files and known-areas, plotting data to the two colour spaces. Testing then prioritises one over the other, before saving the classifier to file. As mentioned in Section 4.5, 24 classifiers are saved in total.

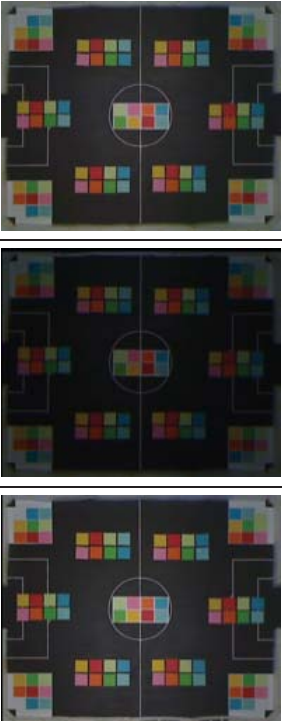
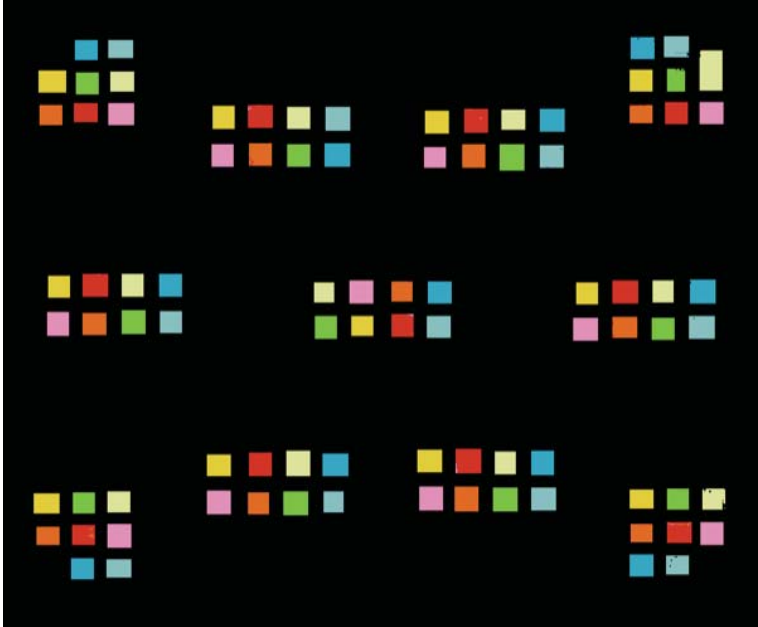


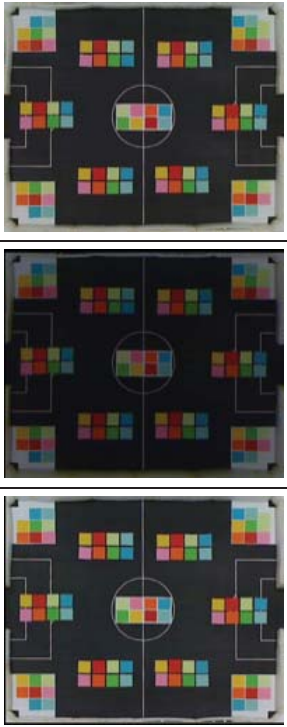
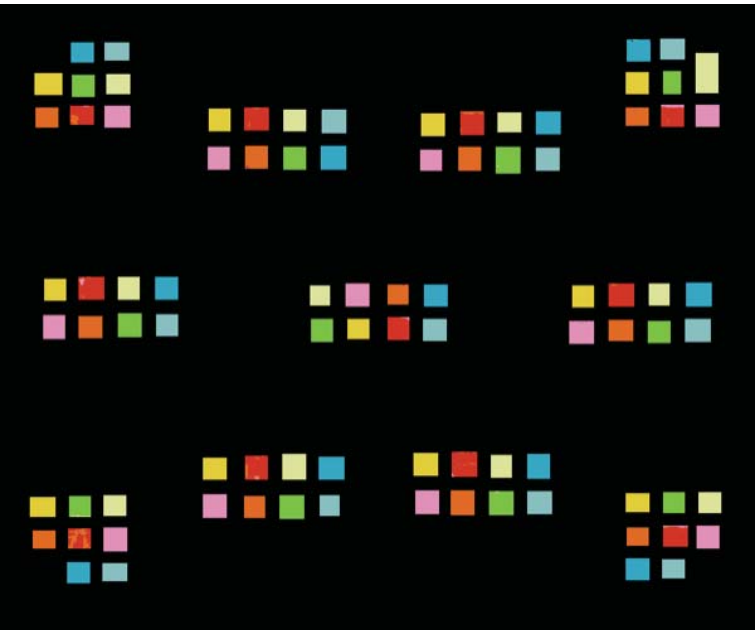
This final program loads each classifier from file, along with a particular lighting scenario. Outputs such as confusion matrices, true positives and false positives are generated here for each scene, saved directly to CSV file for further analysis.

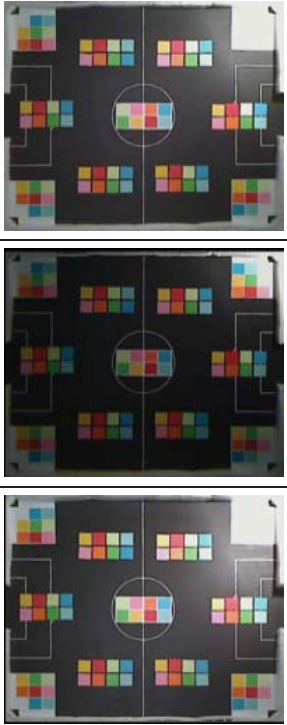
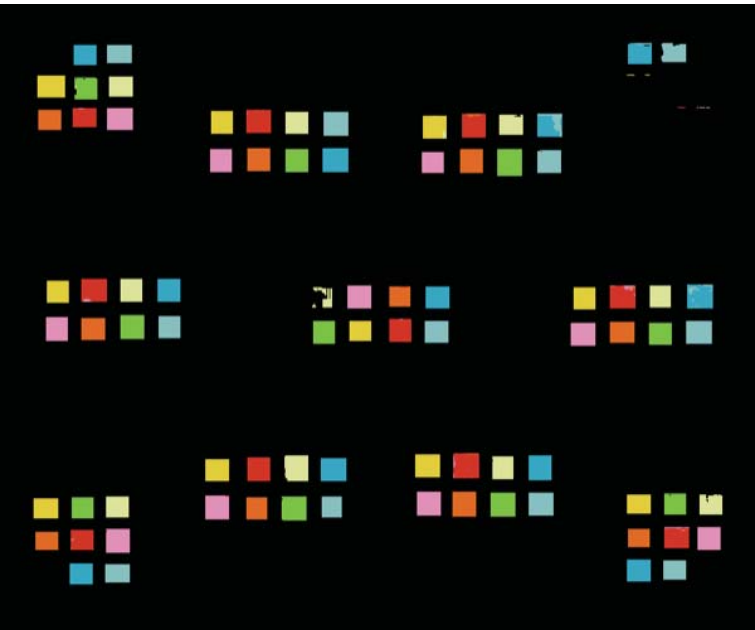
Appendix B: Visual MIDDEC System Results

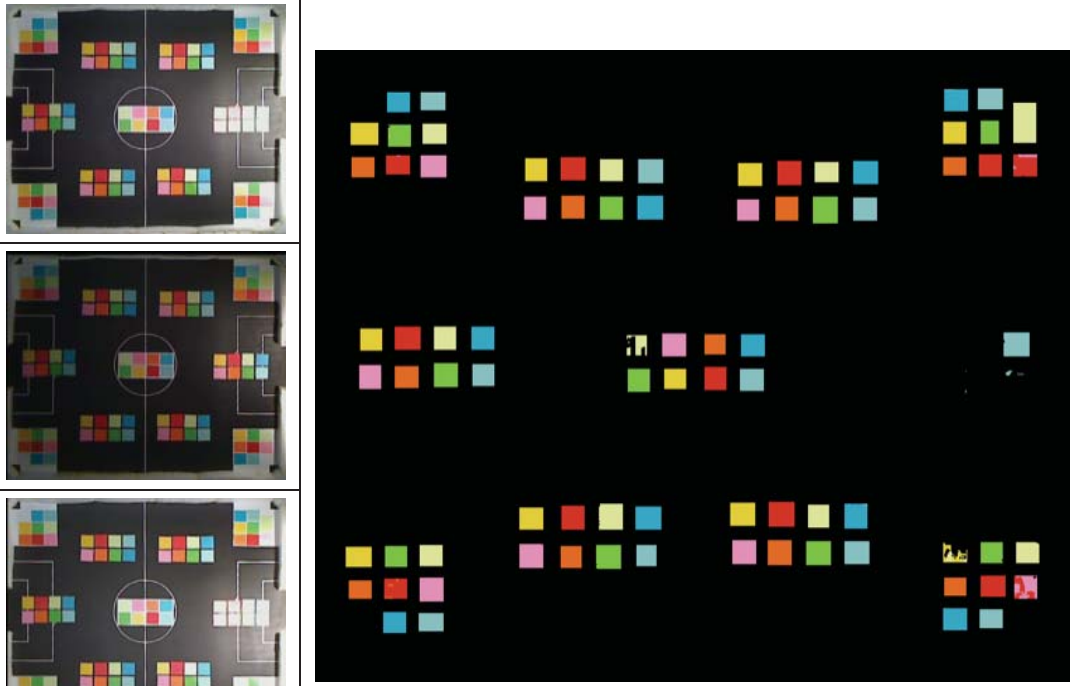
The pages following list all of the results for the MIDECC system, showing inputs and outputs for corresponding lighting scenarios.

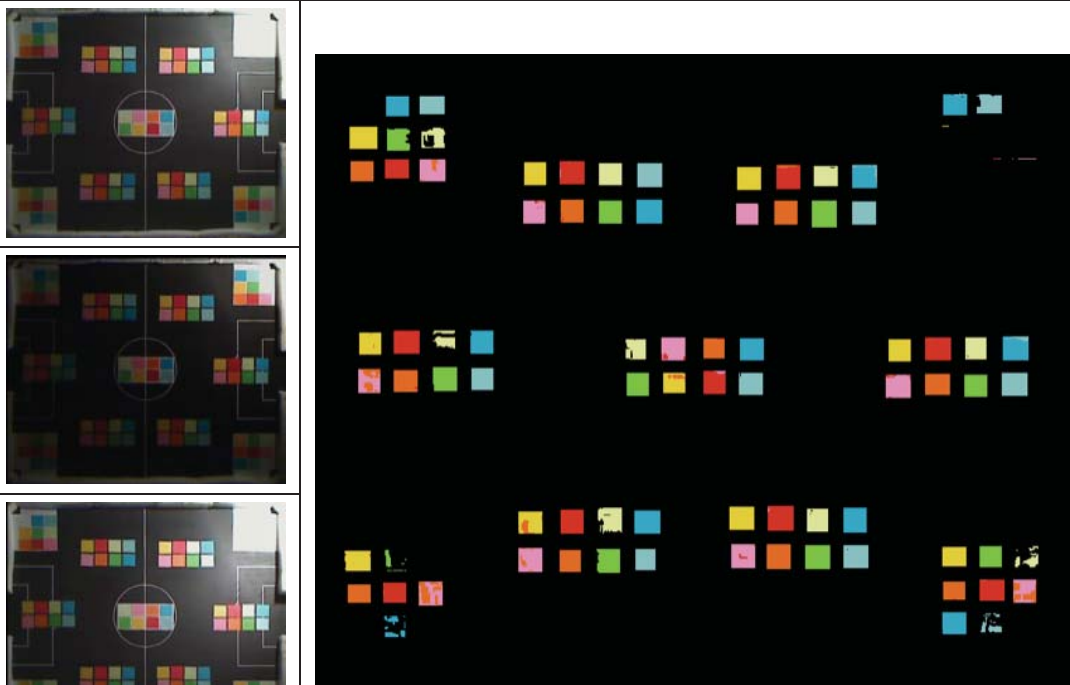
Scenario	Background	Data Set	TP	FP
Even Bright	Known-Areas Only	Training	99.38%	0.47%
 				

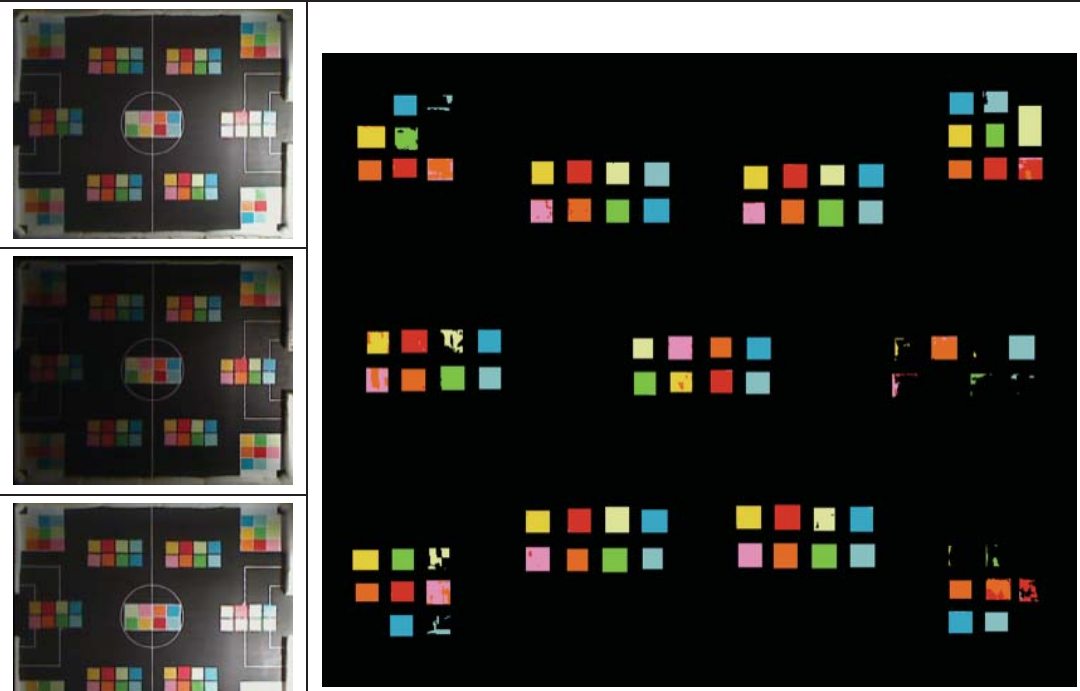
Scenario	Background	Data Set	TP	FP
Even Dim	Known-Areas Only	Training	99.53%	0.34%
 				


Scenario	Background	Data Set	TP	FP
Even Medium	Known-Areas Only	Training	98.61%	1.39%
 				

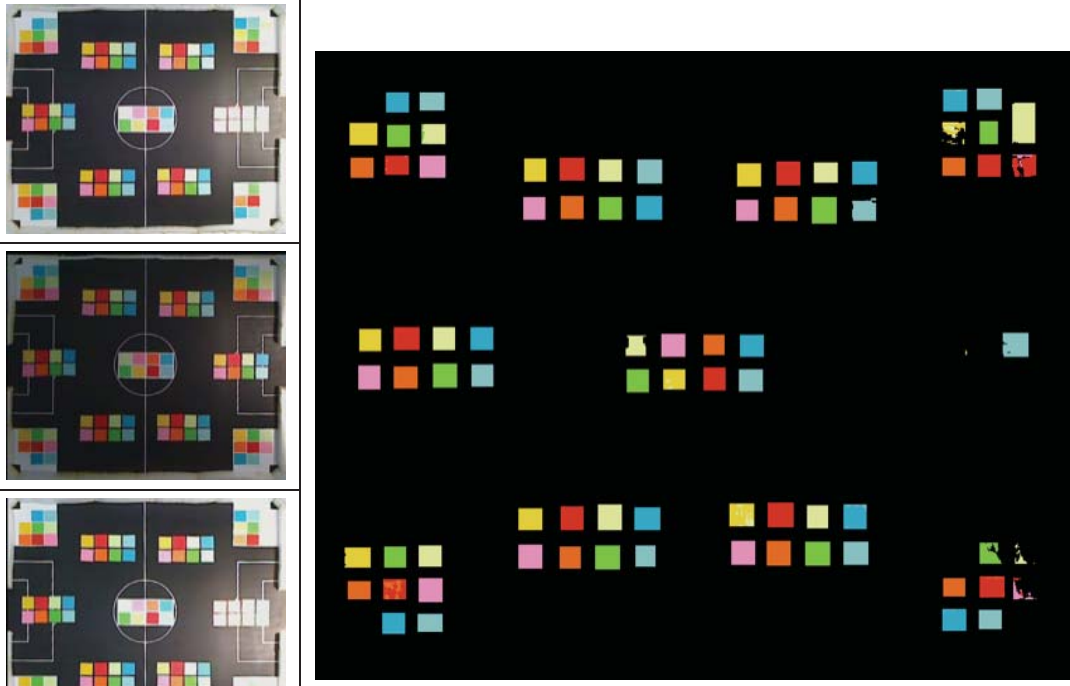
Scenario	Background	Data Set	TP	FP
Harsh Moderate 1	Known-Areas Only	Training	90.83%	1.13%
 				

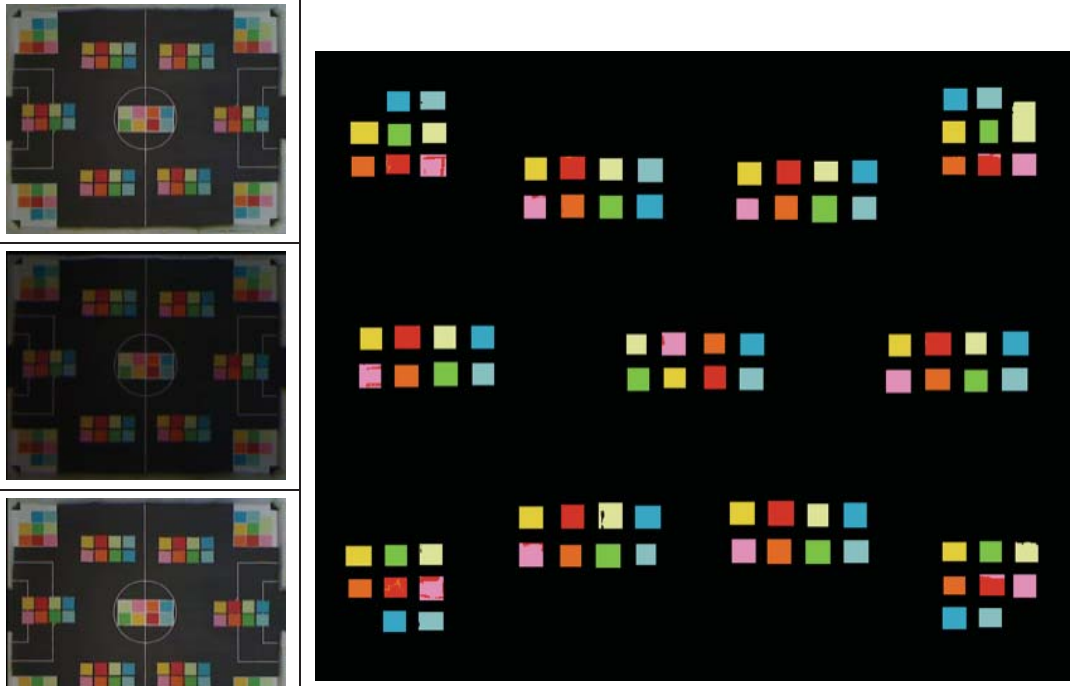
Scenario	Background	Data Set	TP	FP
Harsh Moderate 2	Known-Areas Only	Training	88.05%	3.04%
				

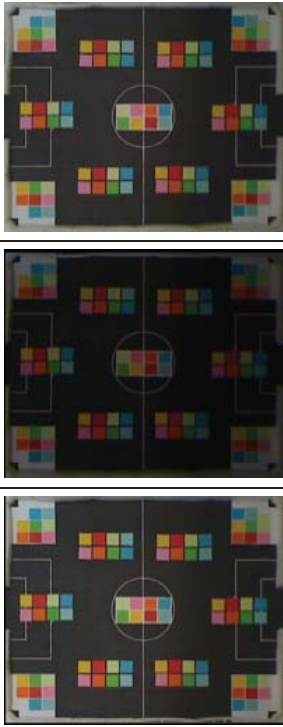
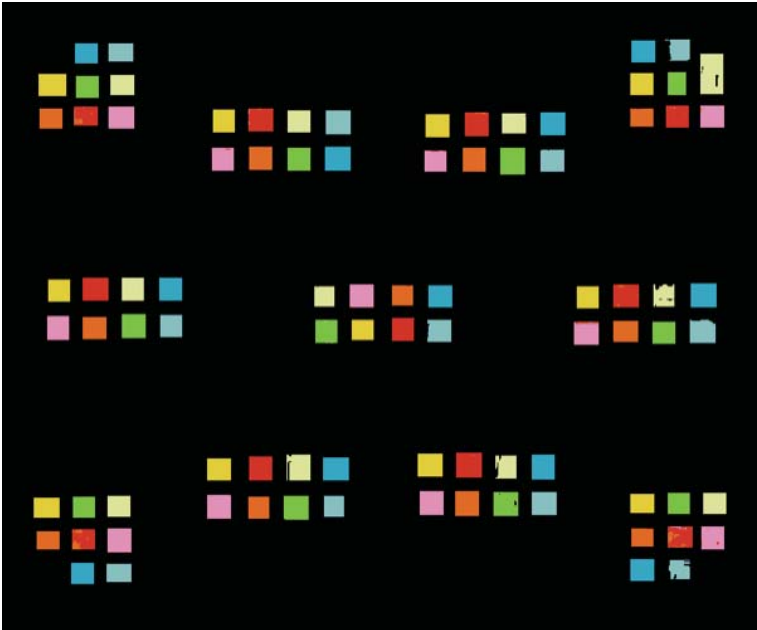
Scenario	Background	Data Set	TP	FP
Harsh Severe 1	Known-Areas Only	Training	83.50%	2.86%
				

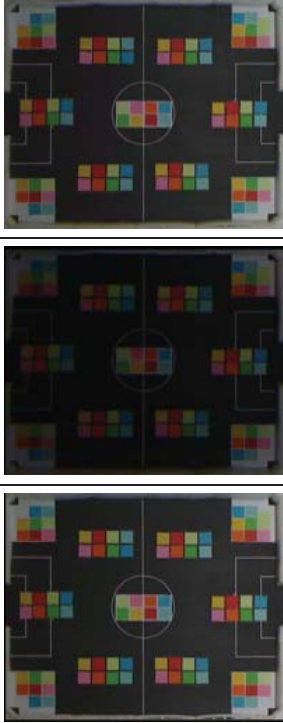
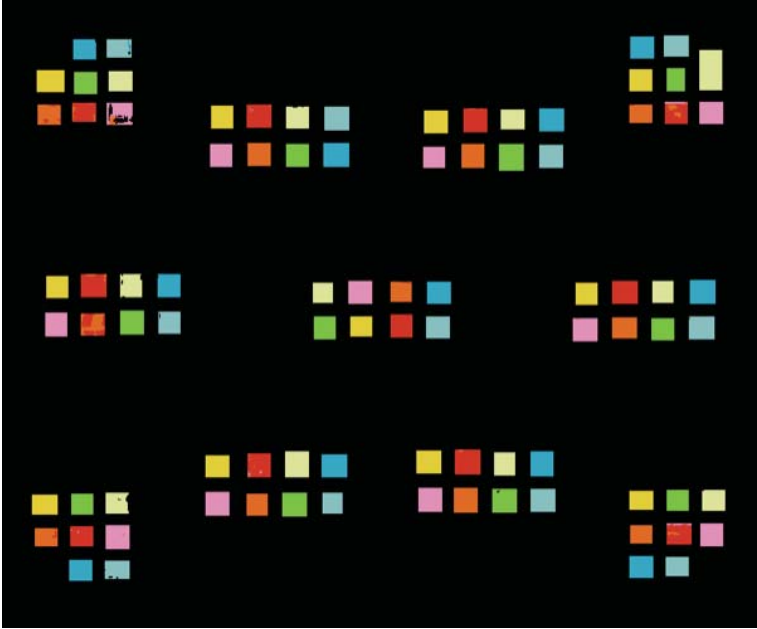
Scenario	Background	Data Set	TP	FP
Harsh Severe 2	Known-Areas Only	Training	79.20%	7.32%
				


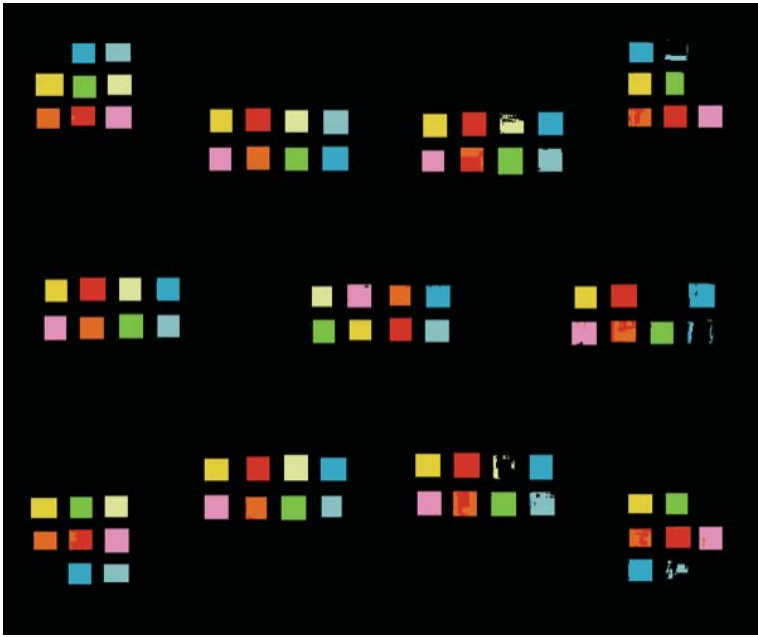
Scenario	Background	Data Set	TP	FP
Harsh Slight 1	Known-Areas Only	Training	88.57%	3.15%
				

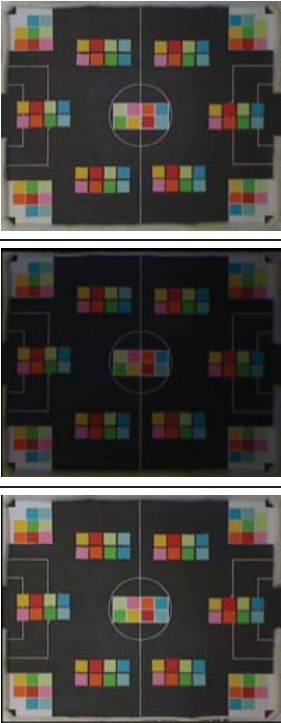
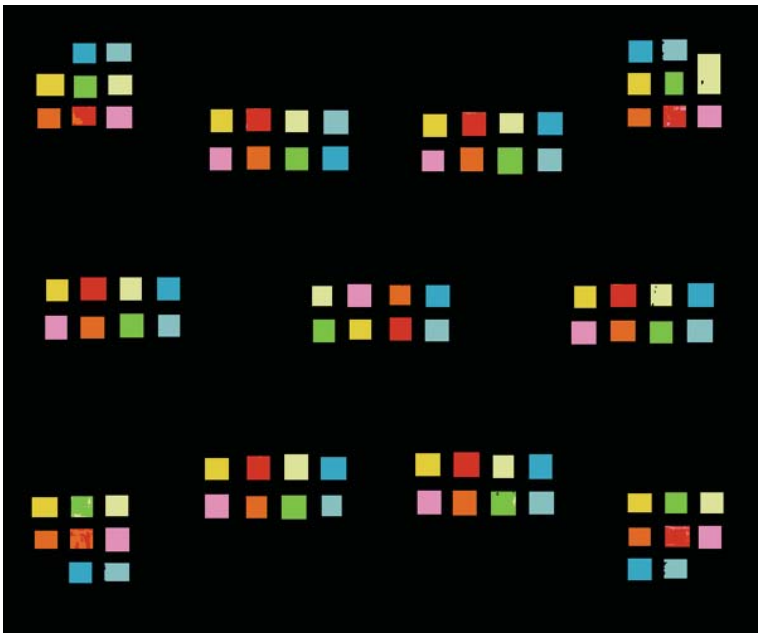
Scenario	Background	Data Set	TP	FP
Harsh Slight 2	Known-Areas Only	Training	84.45%	3.42%
				

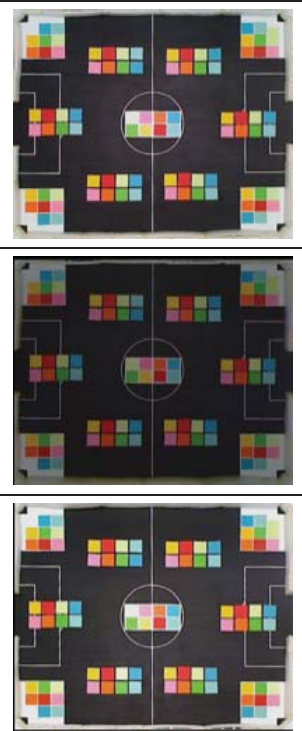
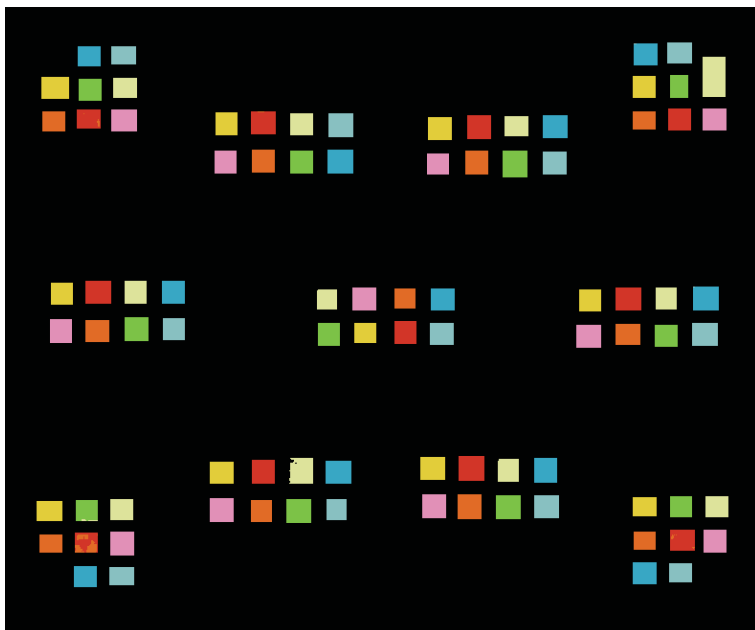
Scenario	Background	Data Set	TP	FP
Uneven Moderate 1	Known-Areas Only	Training	98.41%	1.39%
				

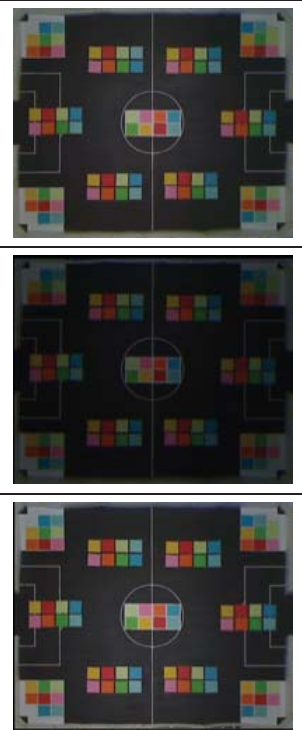
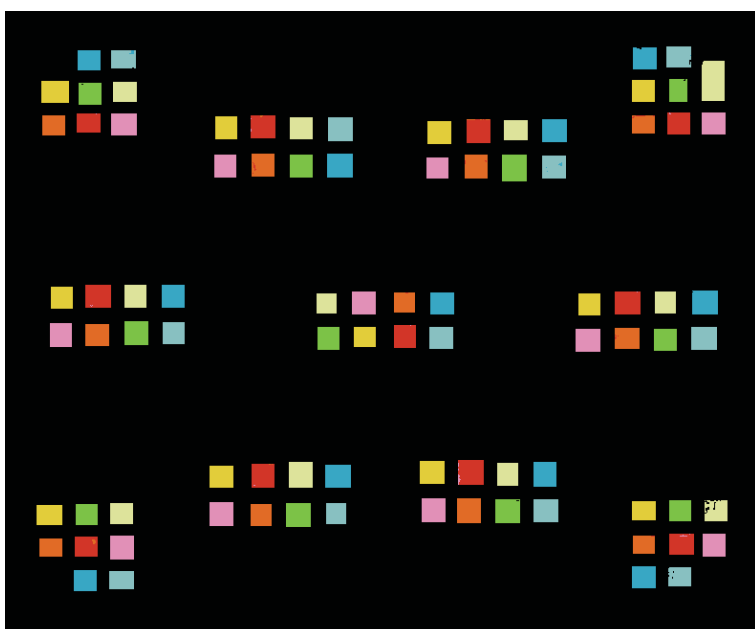
Scenario	Background	Data Set	TP	FP
Uneven Moderate 2	Known-Areas Only	Training	98.19%	0.88%
				

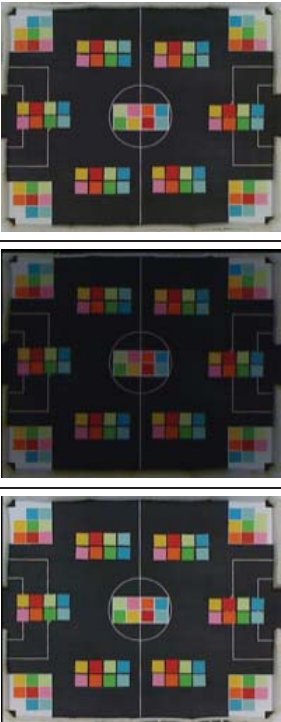
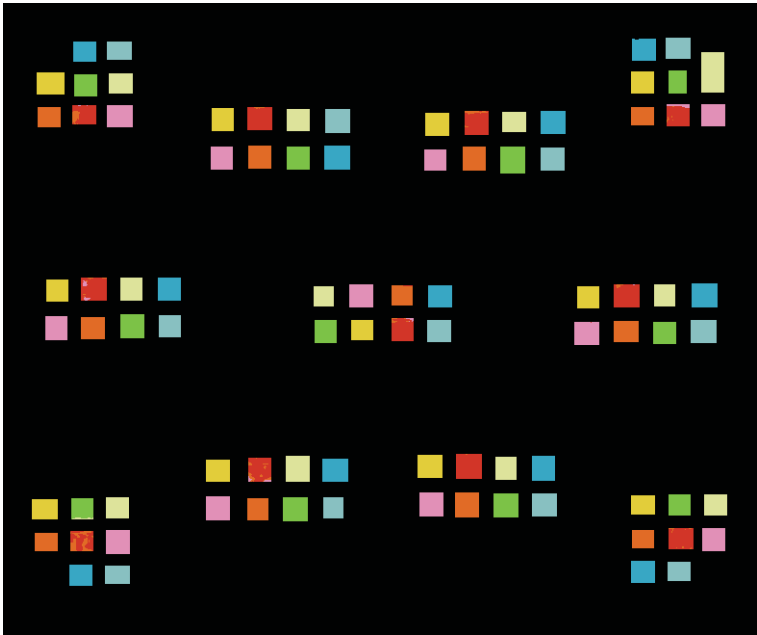
Scenario	Background	Data Set	TP	FP
Uneven Severe 1	Known-Areas Only	Training	98.02%	1.49%
				

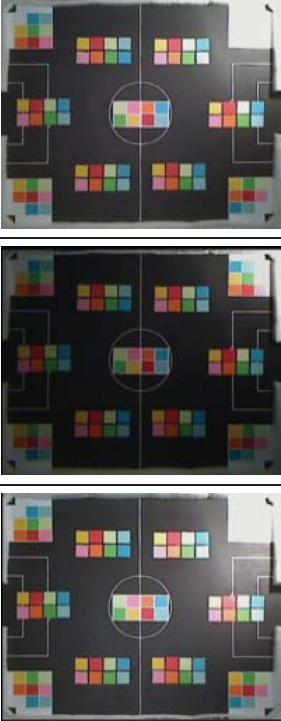
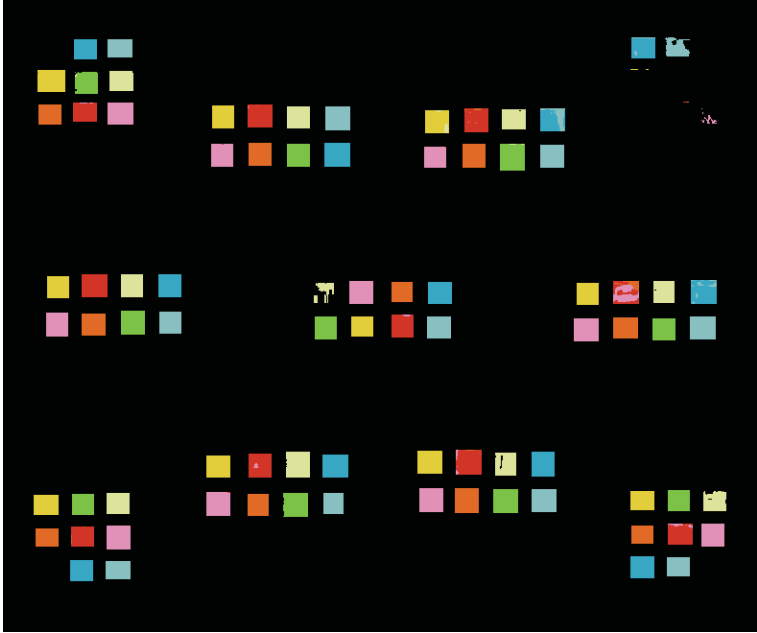
Scenario	Background	Data Set	TP	FP
Uneven Severe 2	Known-Areas Only	Training	88.91%	2.96%
 				


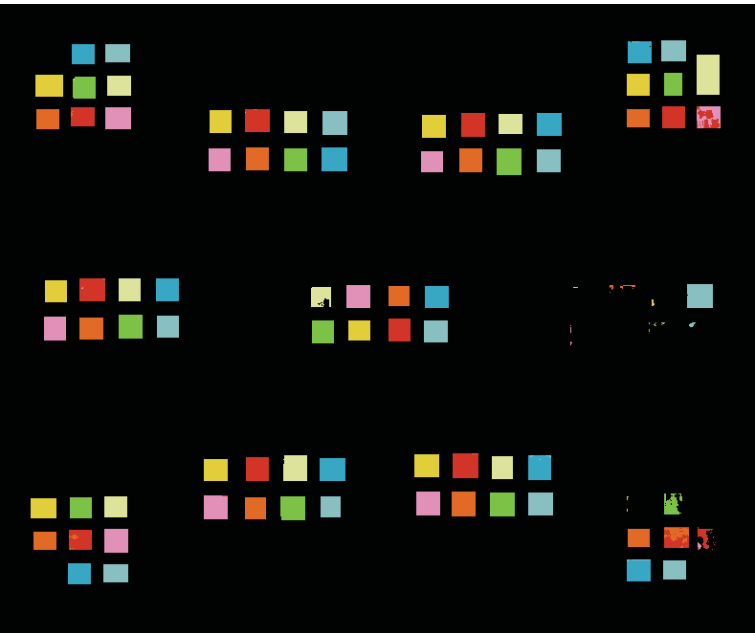
Scenario	Background	Data Set	TP	FP
Uneven Slight	Known-Areas Only	Training	98.61%	1.29%
 				

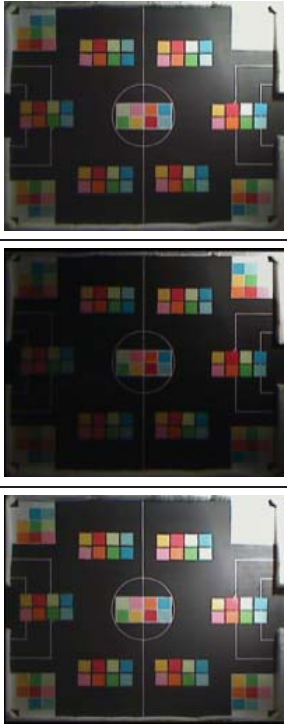
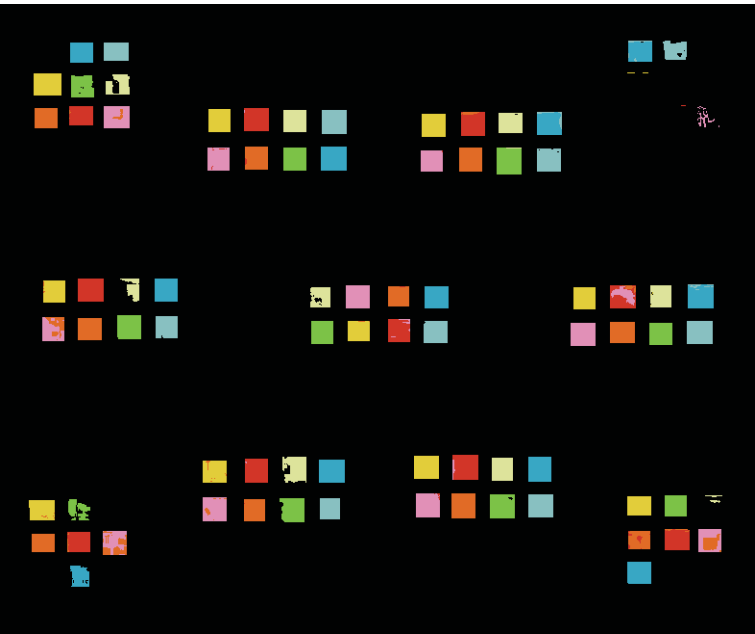
Scenario	Background	Data Set	TP	FP
Even Bright	Known-Areas Only	Test	99.38%	0.55%
 				

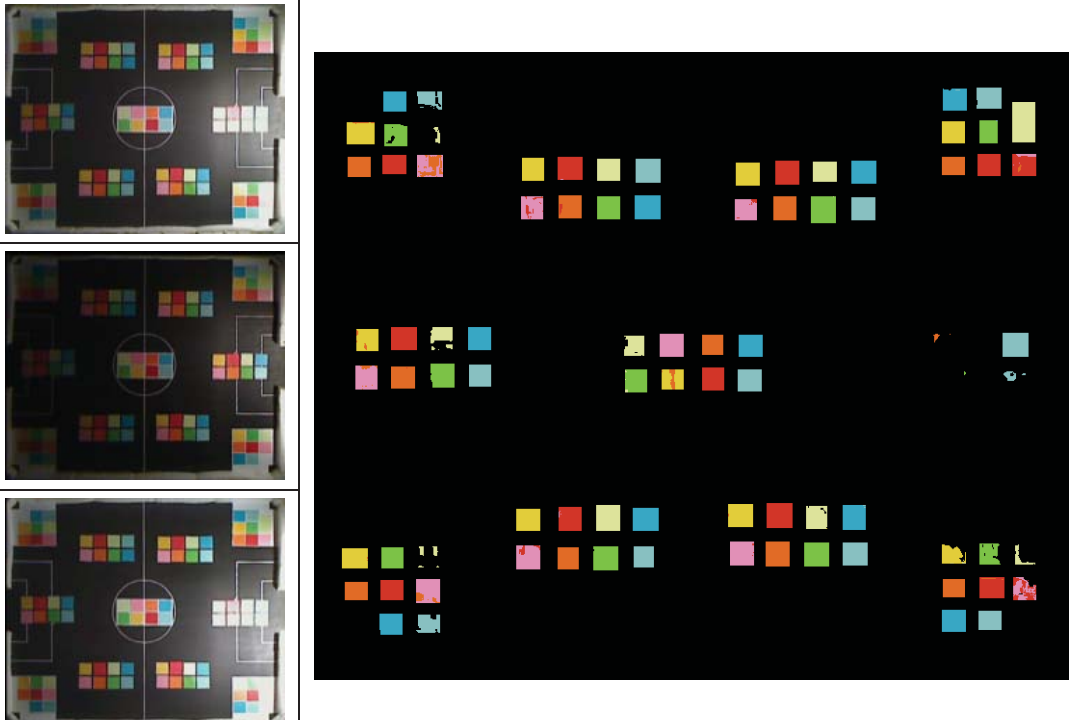
Scenario	Background	Data Set	TP	FP
Even Dim	Known-Areas Only	Test	99.51%	0.33%
 				


Scenario	Background	Data Set	TP	FP
Even Medium	Known-Areas Only	Test	98.56%	1.43%
 				


Scenario	Background	Data Set	TP	FP
Harsh Moderate 1	Known-Areas Only	Test	89.98%	1.86%
 				

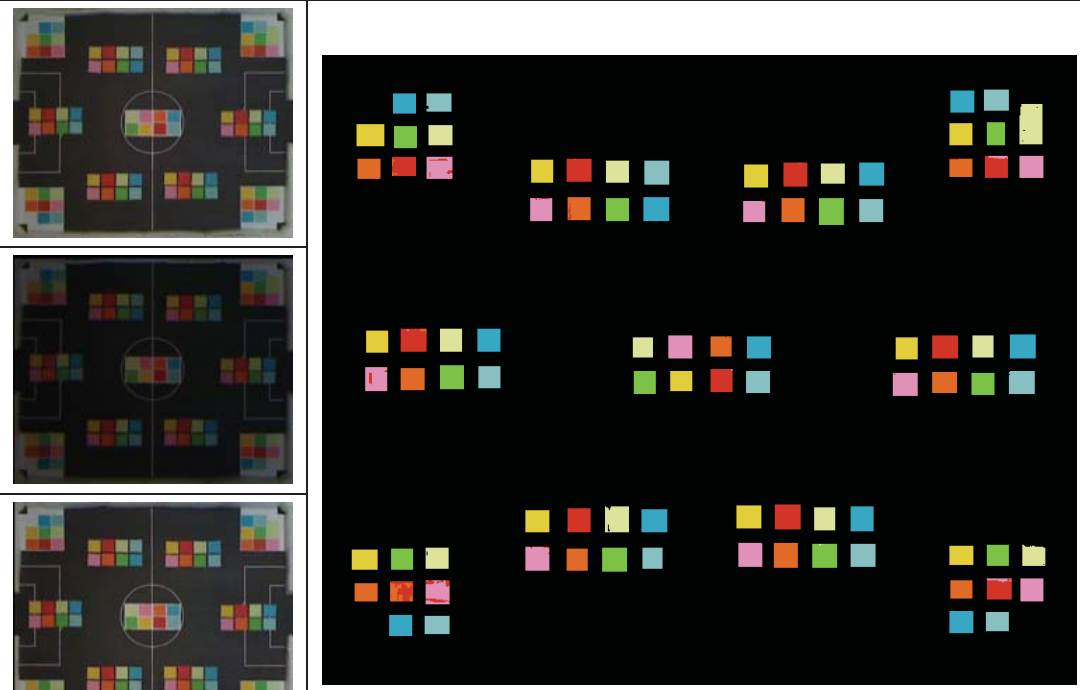
Scenario	Background	Data Set	TP	FP
Harsh Moderate 2	Known-Areas Only	Test	85.73%	3.01%
				

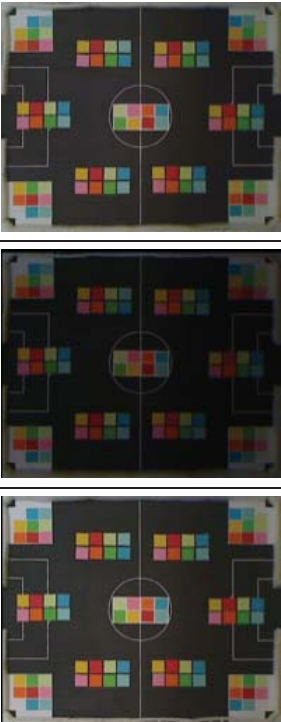
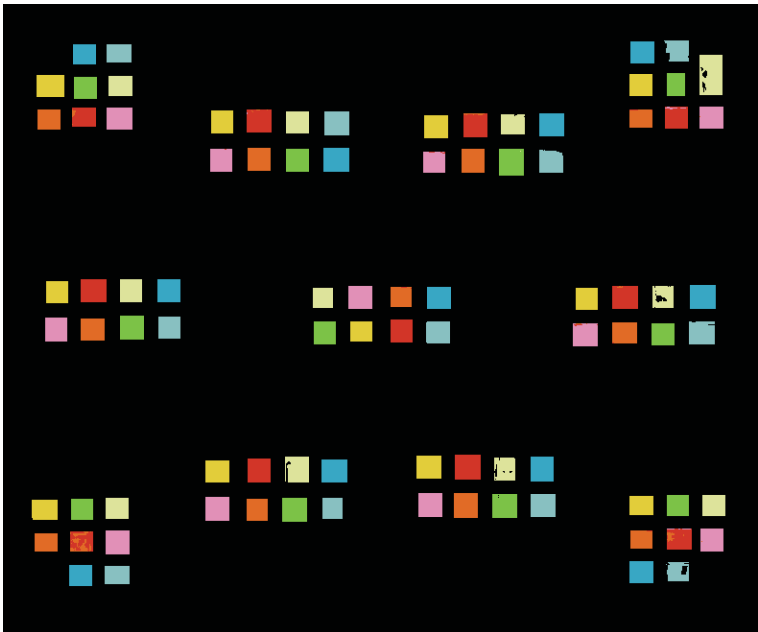
Scenario	Background	Data Set	TP	FP
Harsh Severe 1	Known-Areas Only	Test	83.57%	3.03%
				

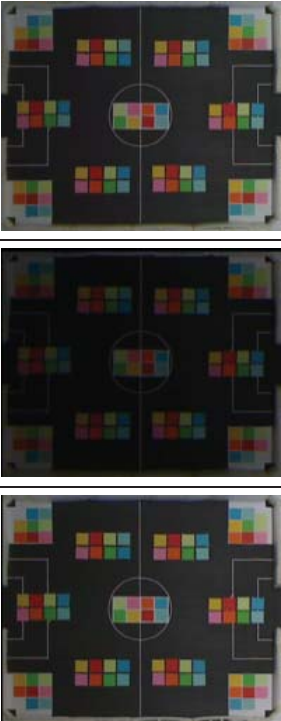
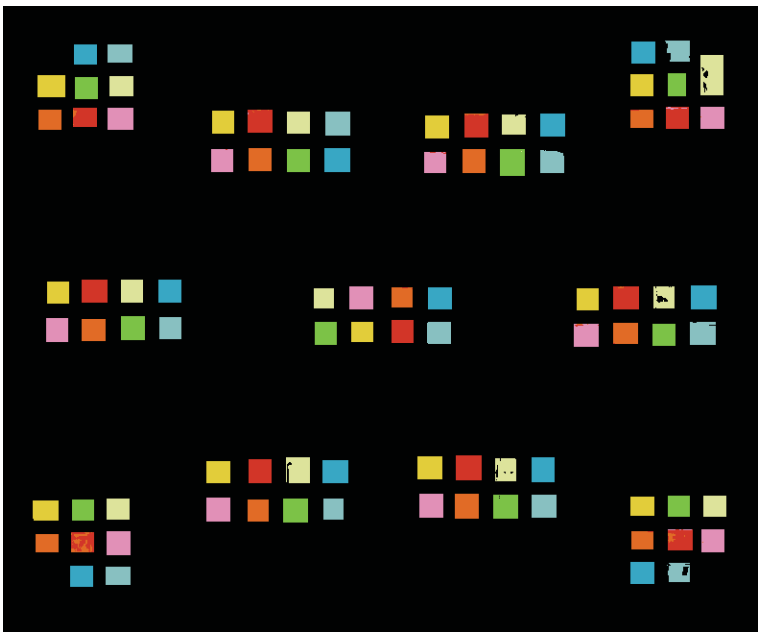
Scenario	Background	Data Set	TP	FP
Harsh Severe 2	Known-Areas Only	Test	83.42%	4.42%
				

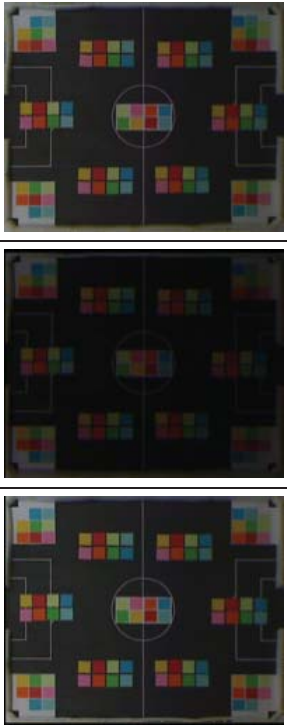
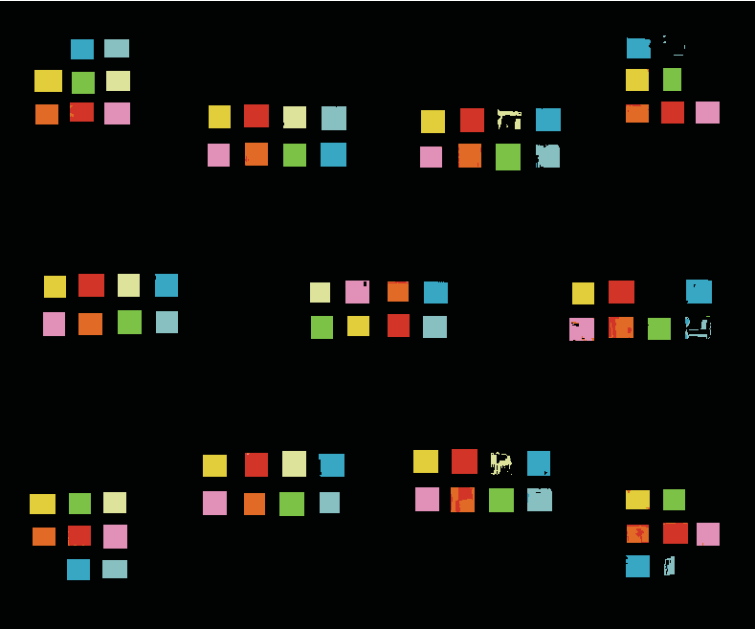
Scenario	Background	Data Set	TP	FP
Harsh Slight 1	Known-Areas Only	Test	89.32%	3.07%
				

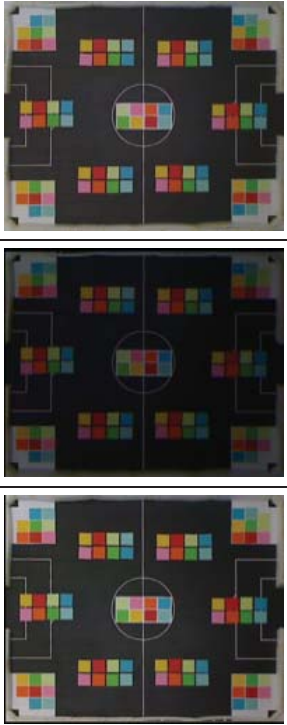
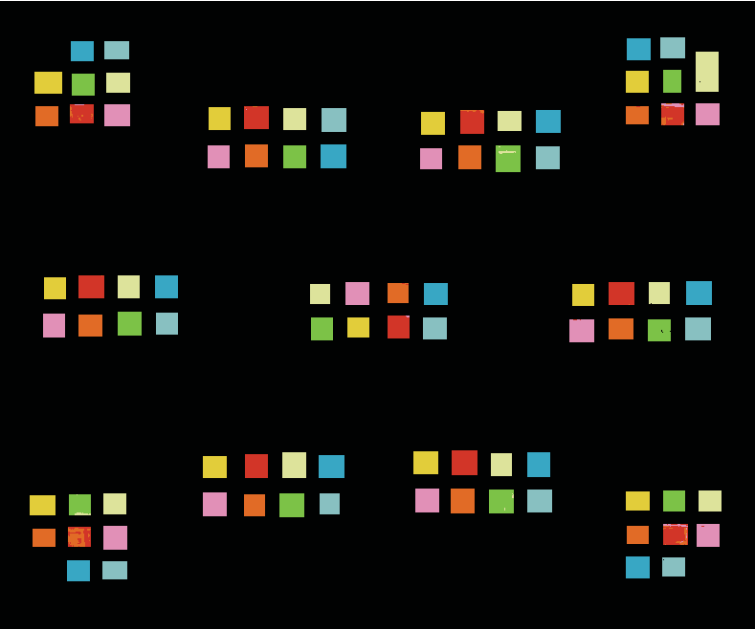
Scenario	Background	Data Set	TP	FP
Harsh Slight 2	Known-Areas Only	Test	83.42%	4.04%
				

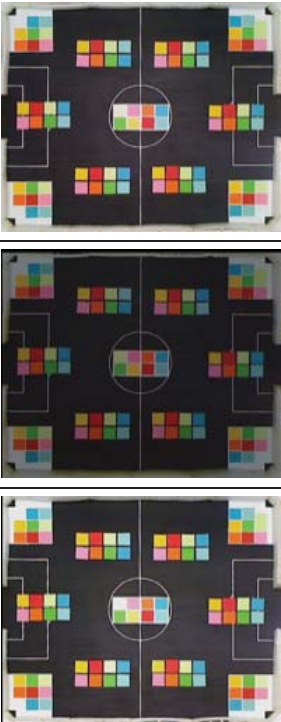

Scenario	Background	Data Set	TP	FP
Uneven Moderate 1	Known-Areas Only	Test	98.45%	1.37%
				

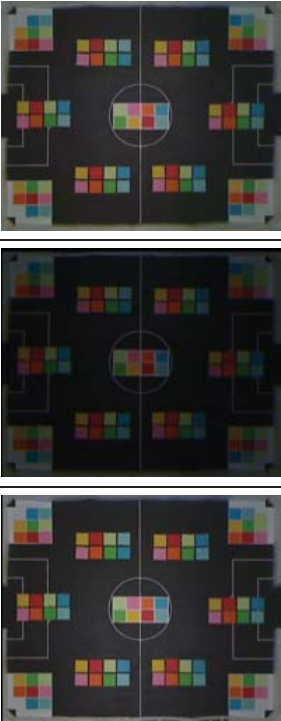
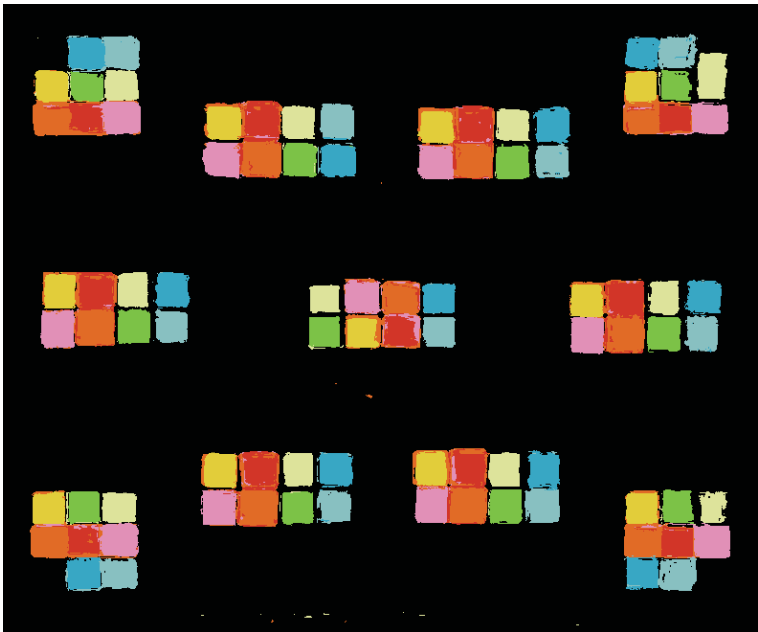
Scenario	Background	Data Set	TP	FP
Uneven Moderate 2	Known-Areas Only	Test	98.34%	0.81%
				

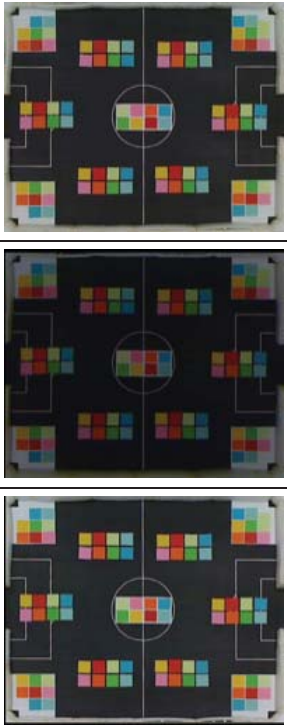
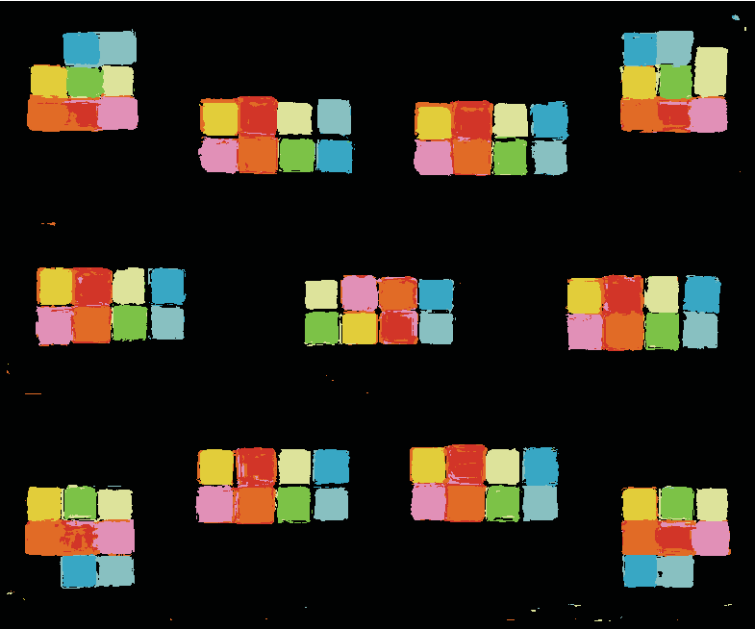
Scenario	Background	Data Set	TP	FP
Uneven Severe 1	Known-Areas Only	Test	97.92%	1.67%
				

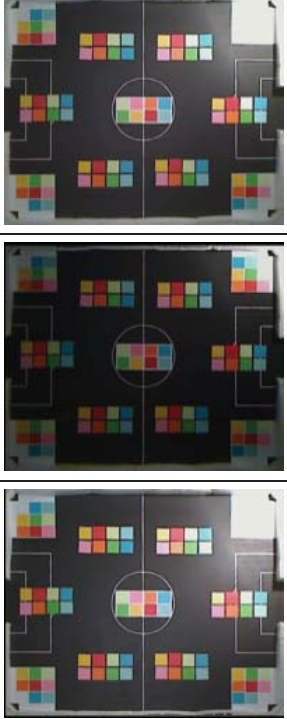
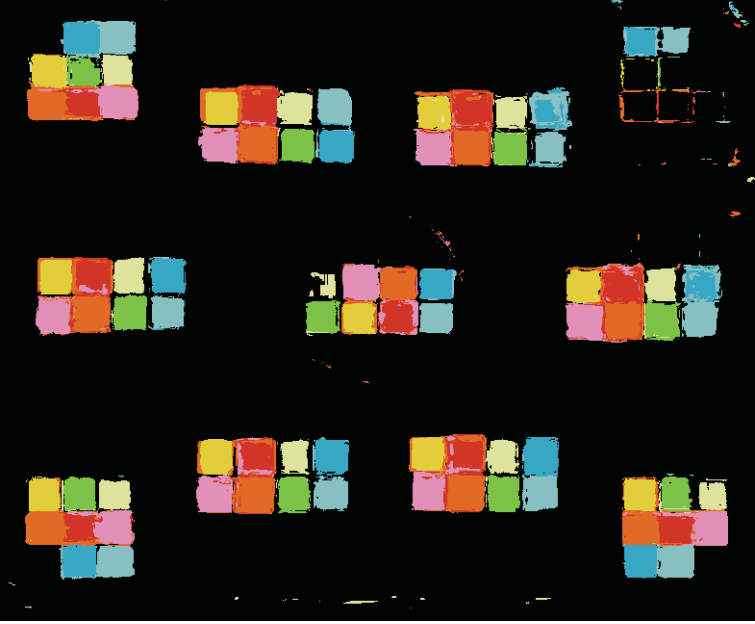
Scenario	Background	Data Set	TP	FP
Uneven Severe 2	Known-Areas Only	Test	90.07%	1.61%
				

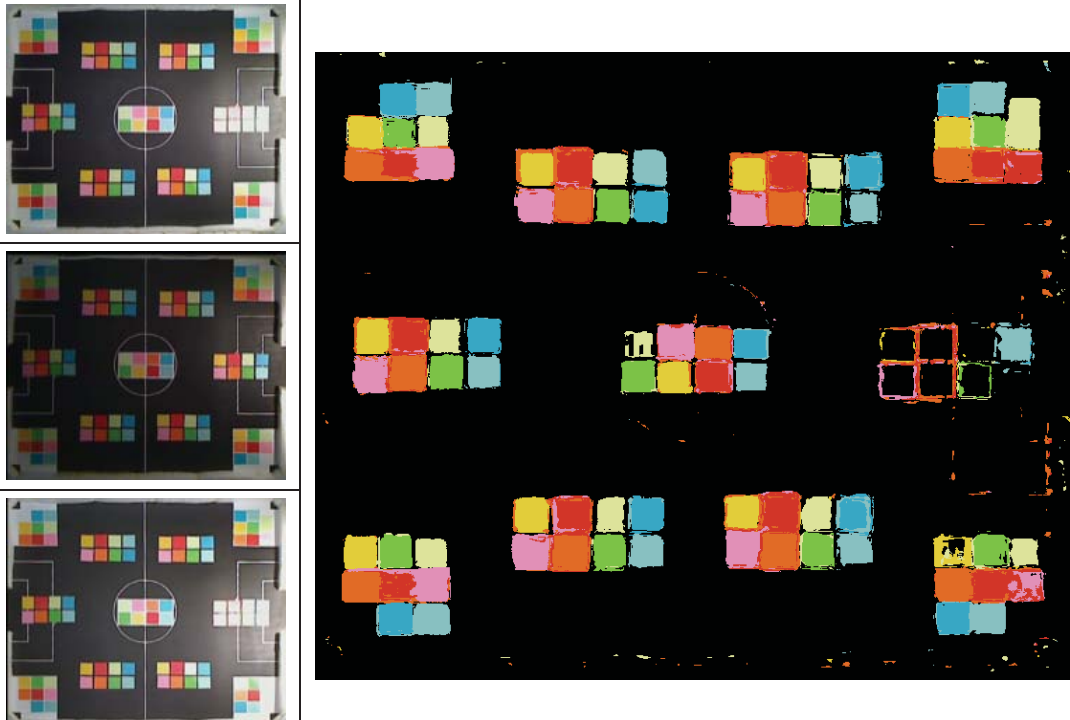
Scenario	Background	Data Set	TP	FP
Uneven Slight	Known-Areas Only	Test	98.48%	1.51%
				

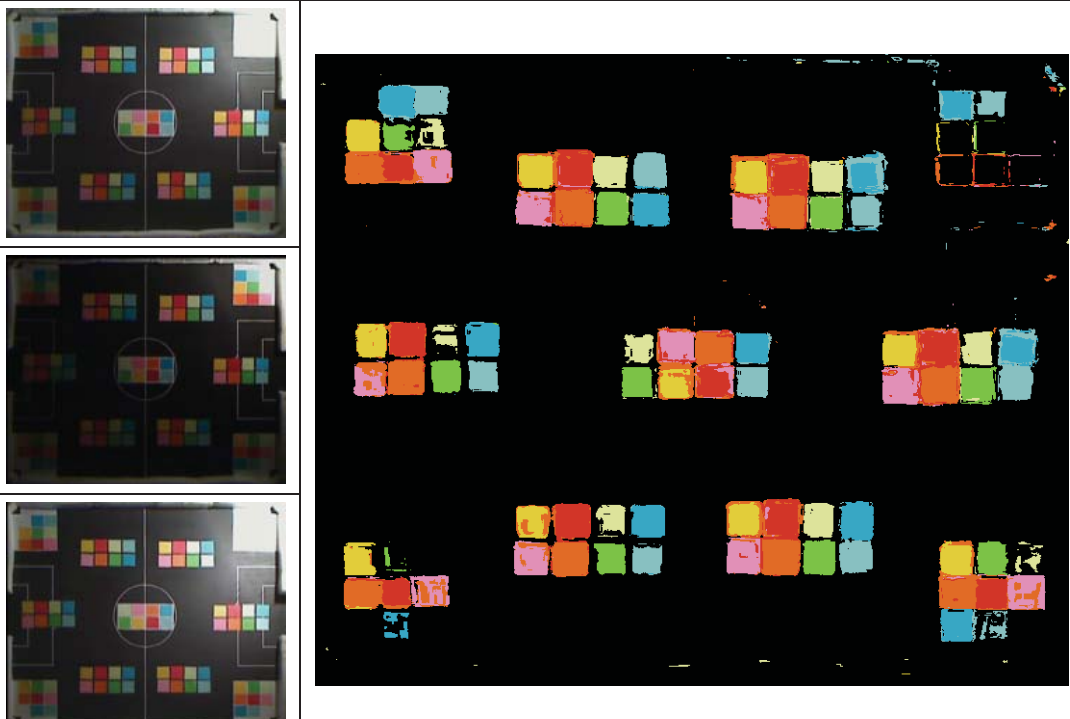
Scenario	Background	Data Set	TP	FP
Even Bright	Background Included	Training	99.38%	3.53%
				

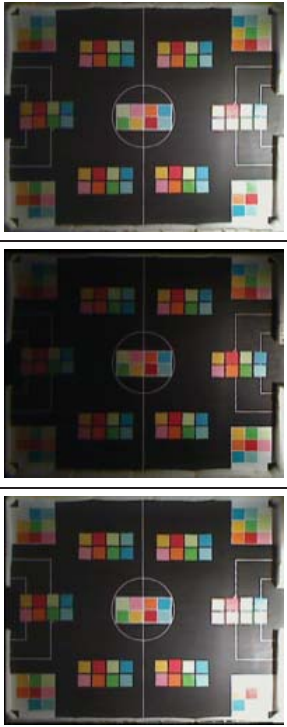
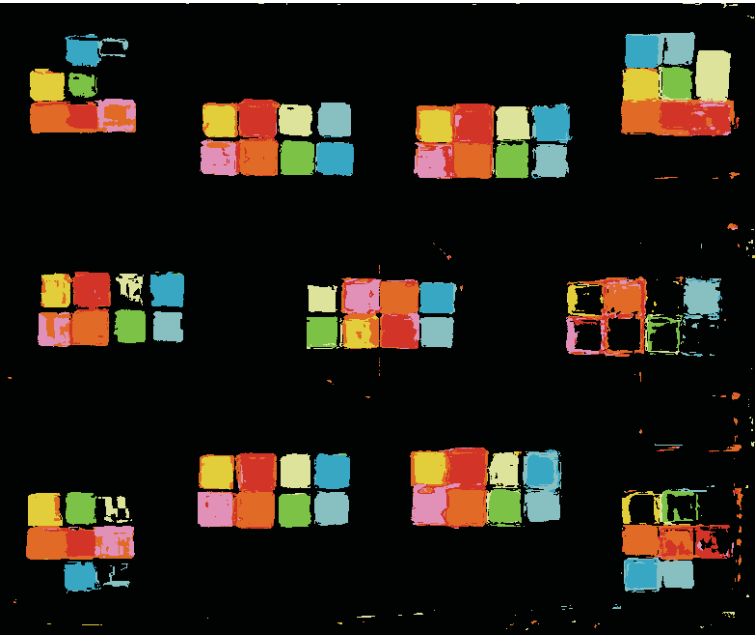
Scenario	Background	Data Set	TP	FP
Even Dim	Background Included	Training	99.53%	0.74%
				

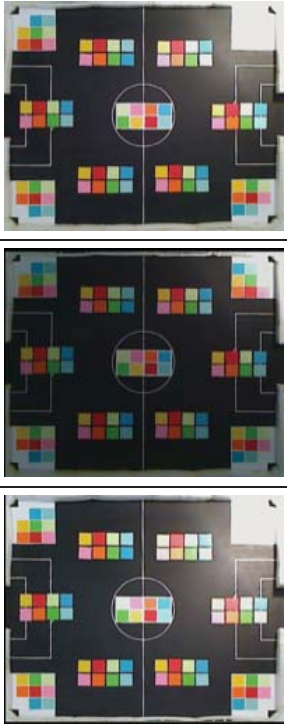
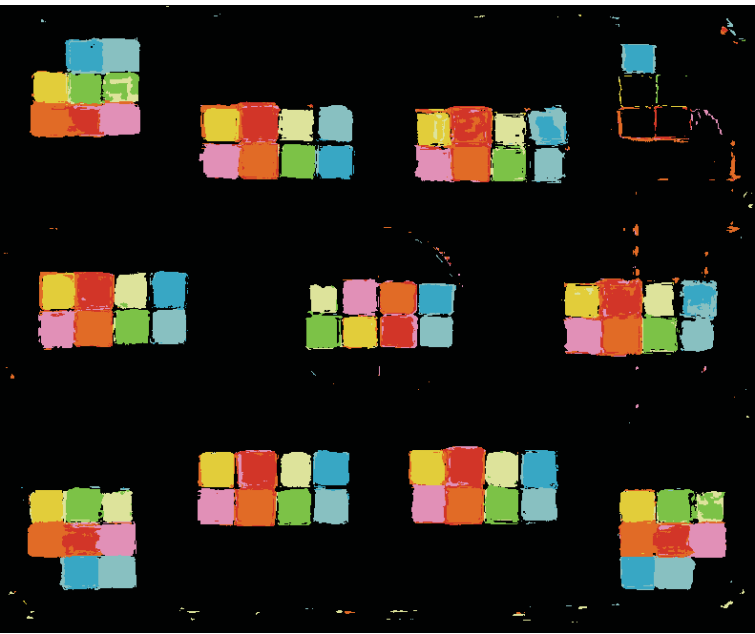
Scenario	Background	Data Set	TP	FP
Even Medium	Background Included	Training	98.61%	2.19%
 				

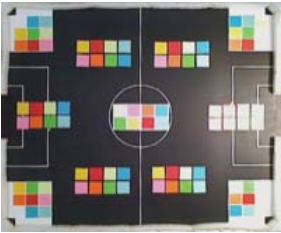
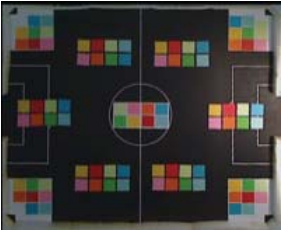

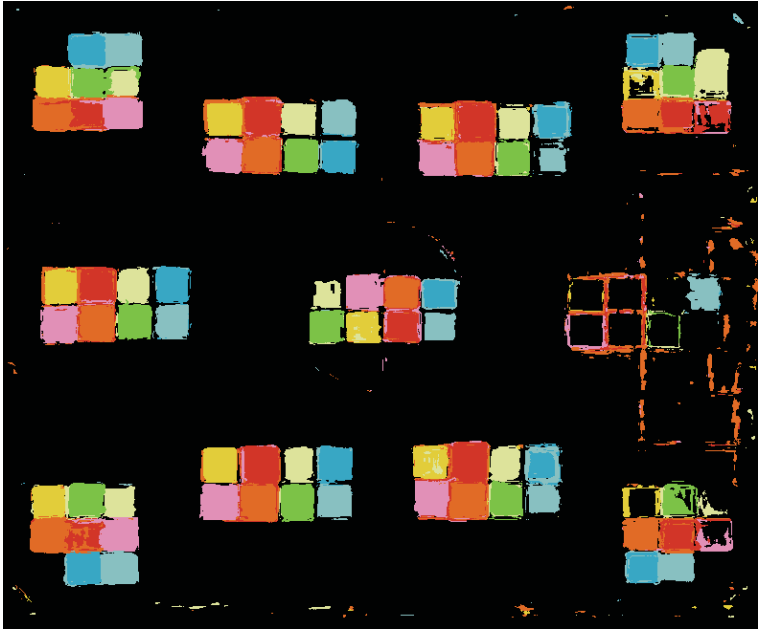
Scenario	Background	Data Set	TP	FP
Harsh Moderate 1	Background Included	Training	90.83%	3.26%
 				


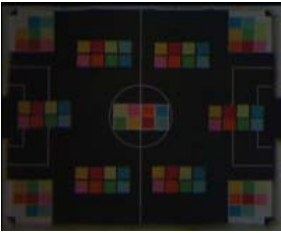
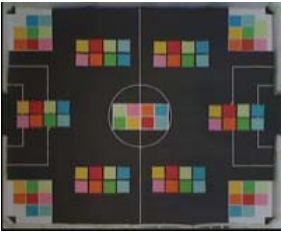
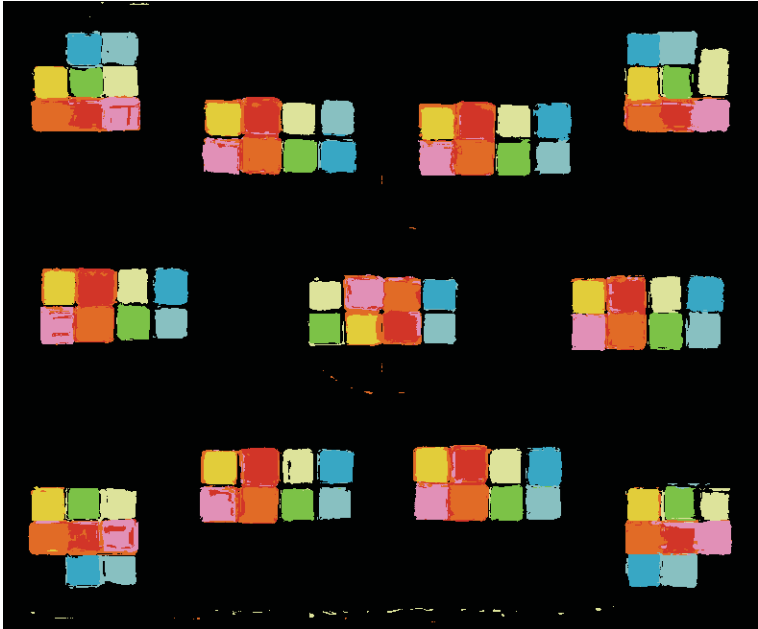
Scenario	Background	Data Set	TP	FP
Harsh Moderate 2	Background Included	Training	88.05%	6.45%
				

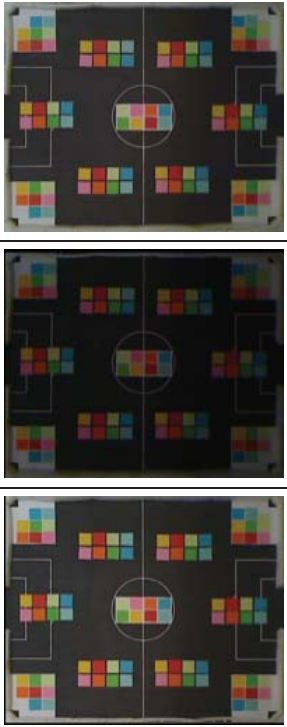
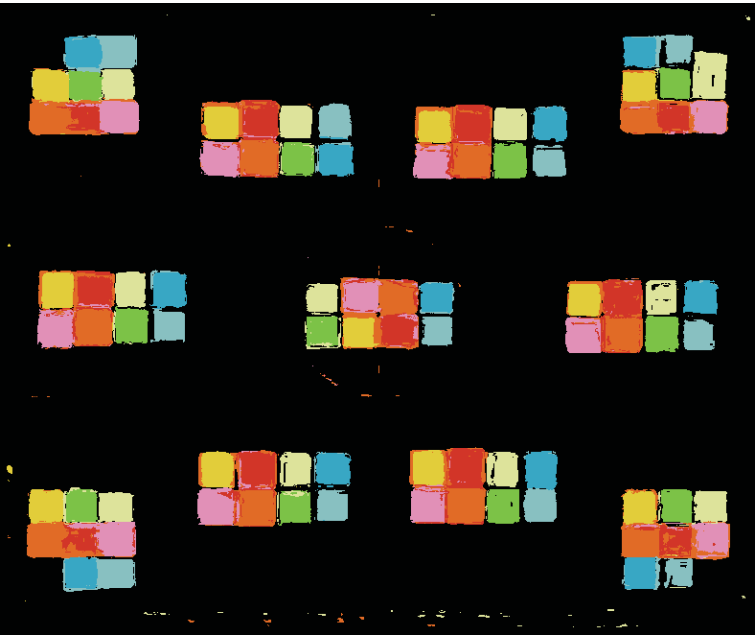
Scenario	Background	Data Set	TP	FP
Harsh Severe 1	Background Included	Training	83.50%	5.05%
				

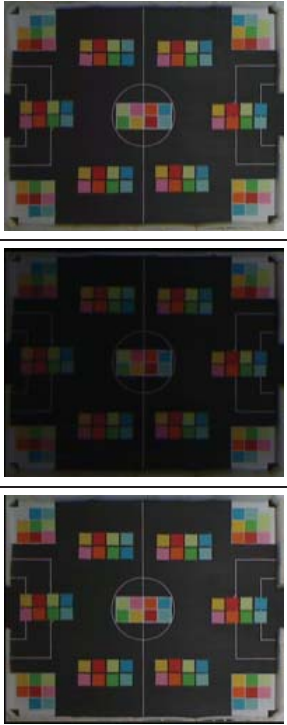
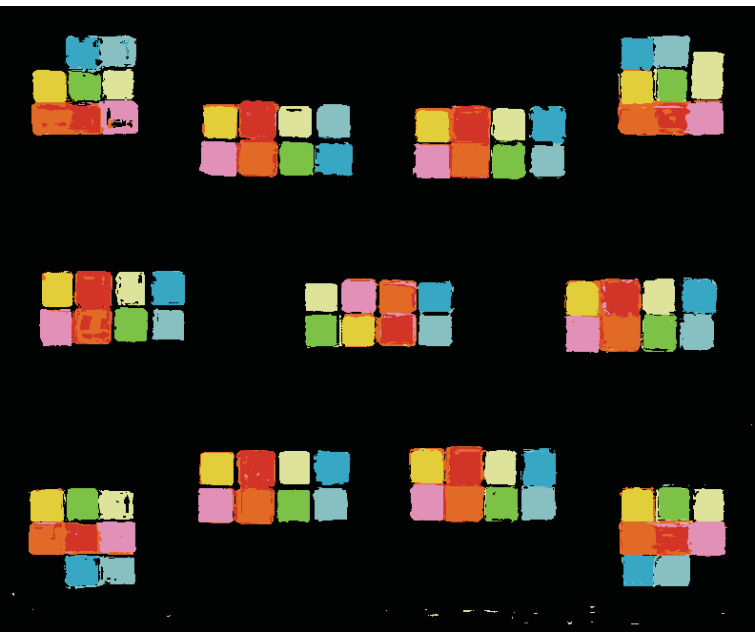
Scenario	Background	Data Set	TP	FP
Harsh Severe 2	Background Included	Training	79.20%	10.51%
 				

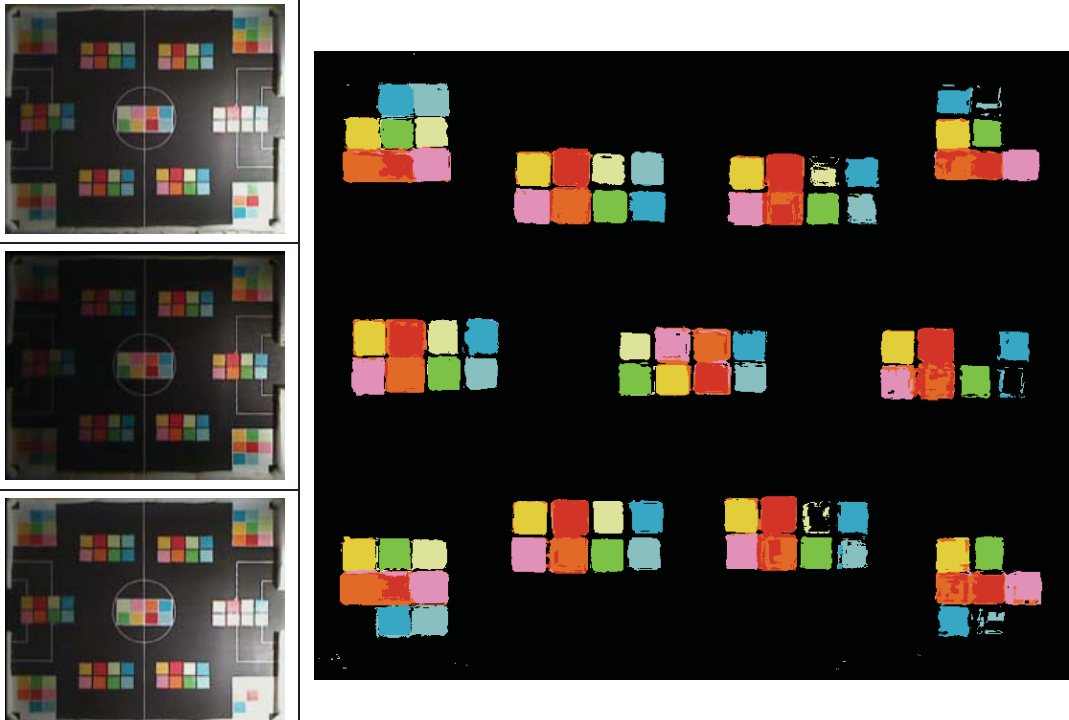
Scenario	Background	Data Set	TP	FP
Harsh Slight 1	Background Included	Training	88.57%	6.55%
 				

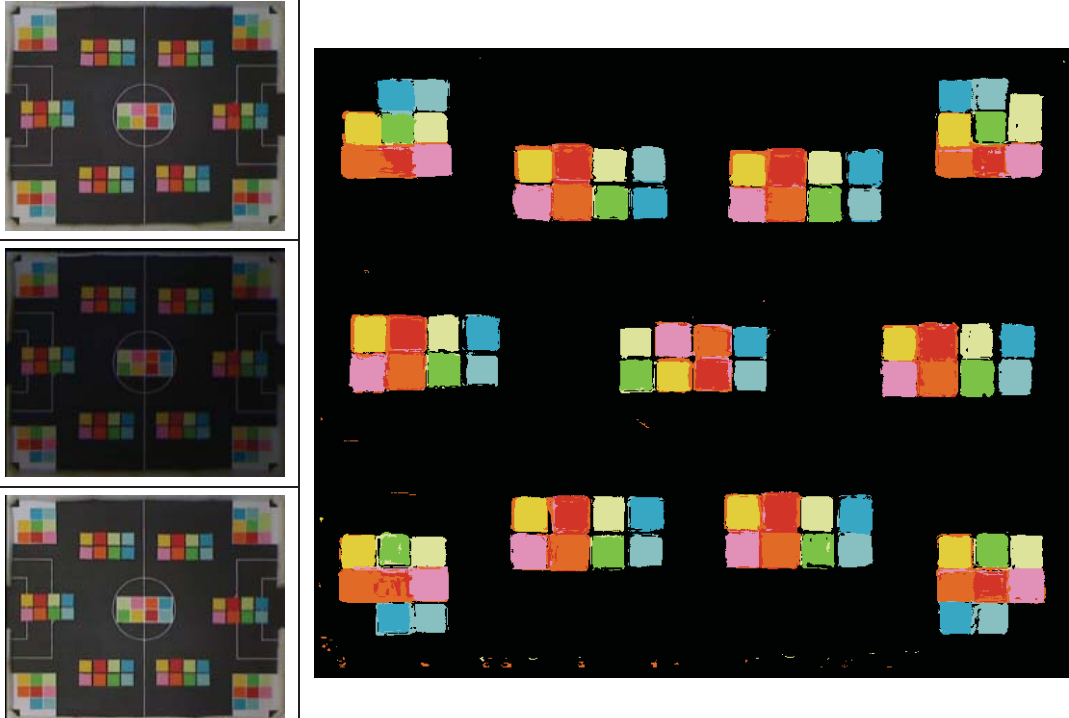
Scenario	Background	Data Set	TP	FP
Harsh Slight 2	Background Included	Training	84.45%	11.03%
  				

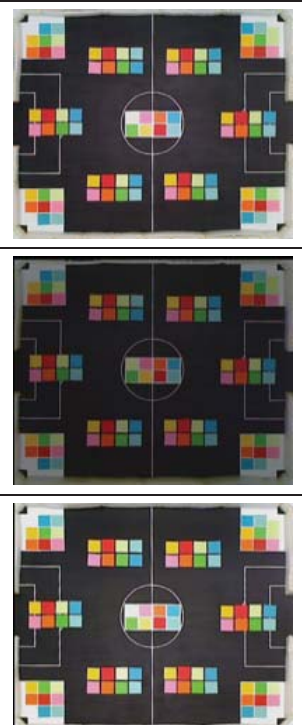
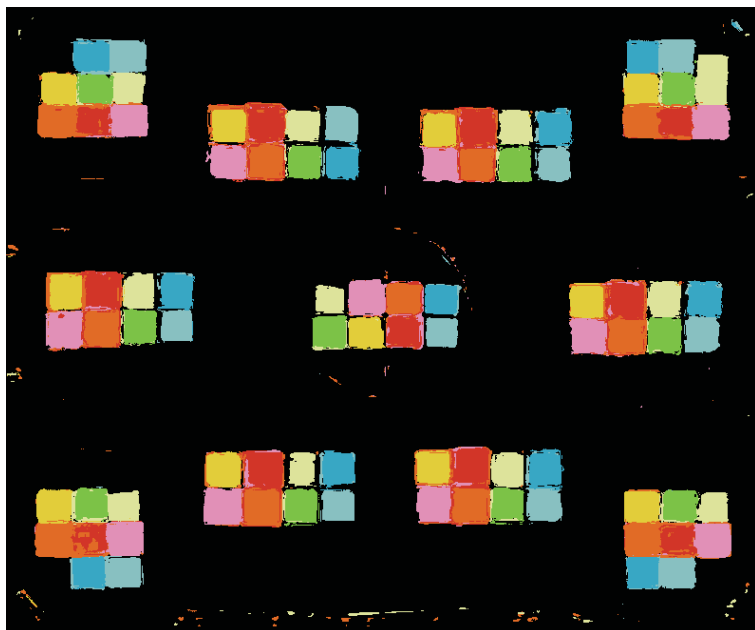
Scenario	Background	Data Set	TP	FP
Uneven Moderate 1	Background Included	Training	98.41%	2.33%
  				

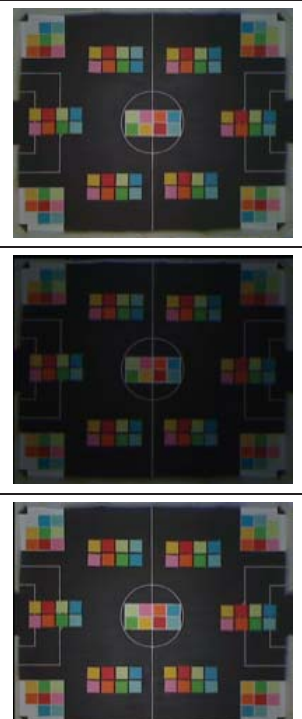
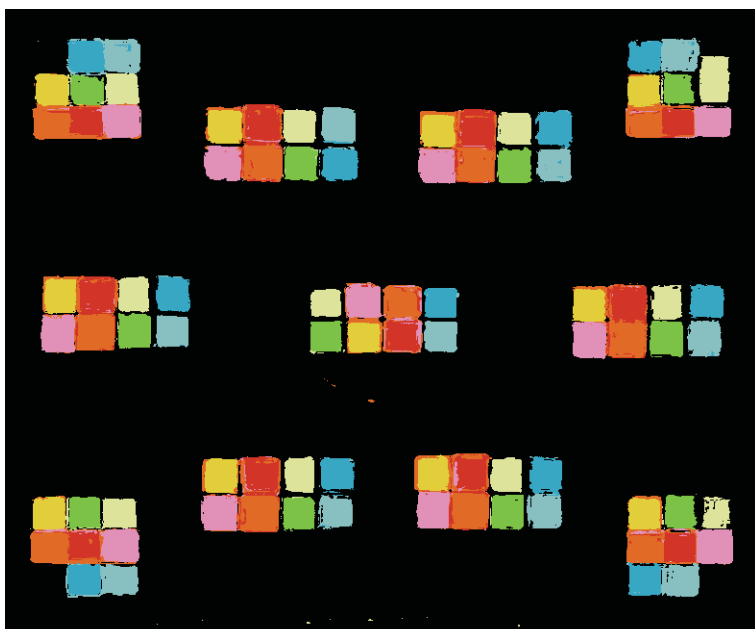
Scenario	Background	Data Set	TP	FP
Uneven Moderate 2	Background Included	Training	98.19%	2.36%
 				

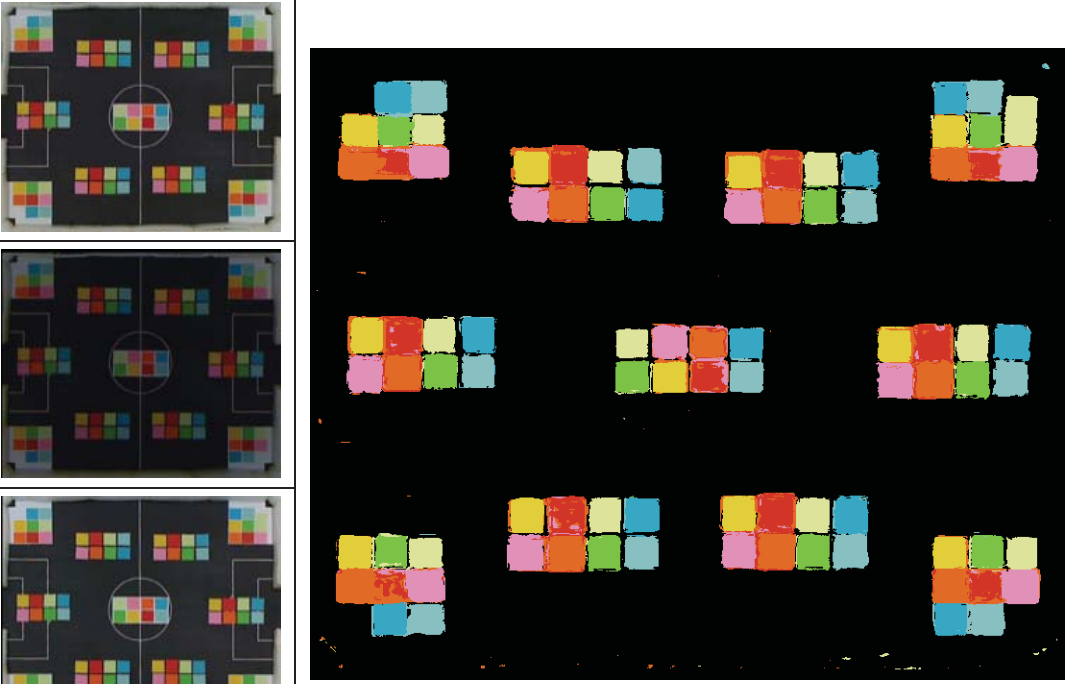
Scenario	Background	Data Set	TP	FP
Uneven Severe 1	Background Included	Training	98.02%	1.93%
 				

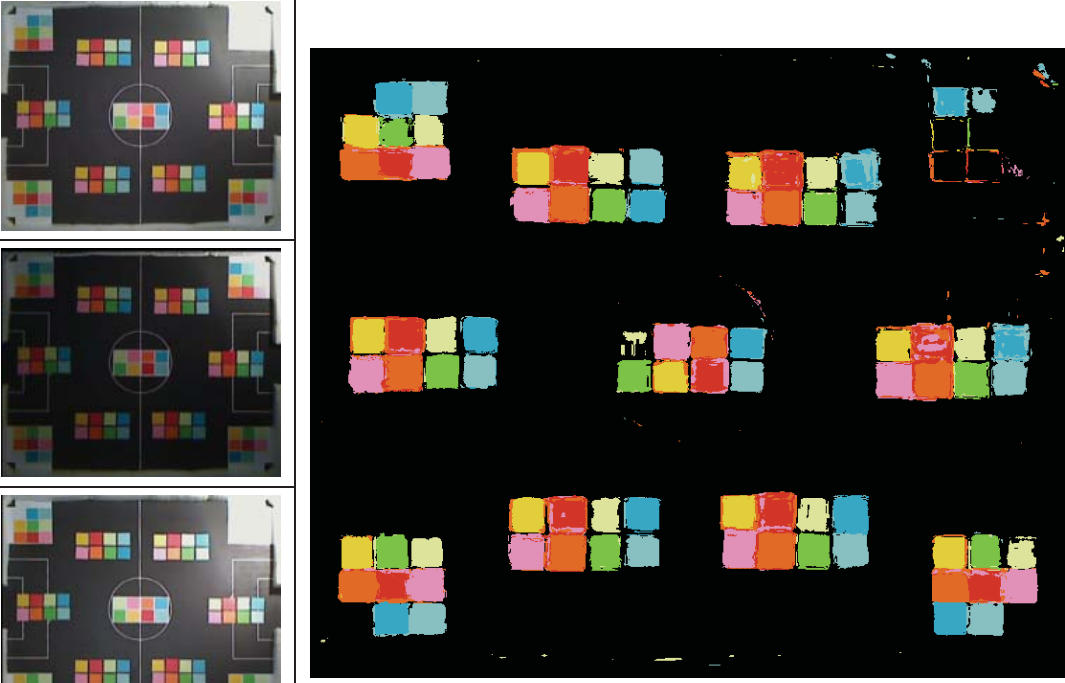
Scenario	Background	Data Set	TP	FP
Uneven Severe 2	Background Included	Training	88.91%	3.14%
				


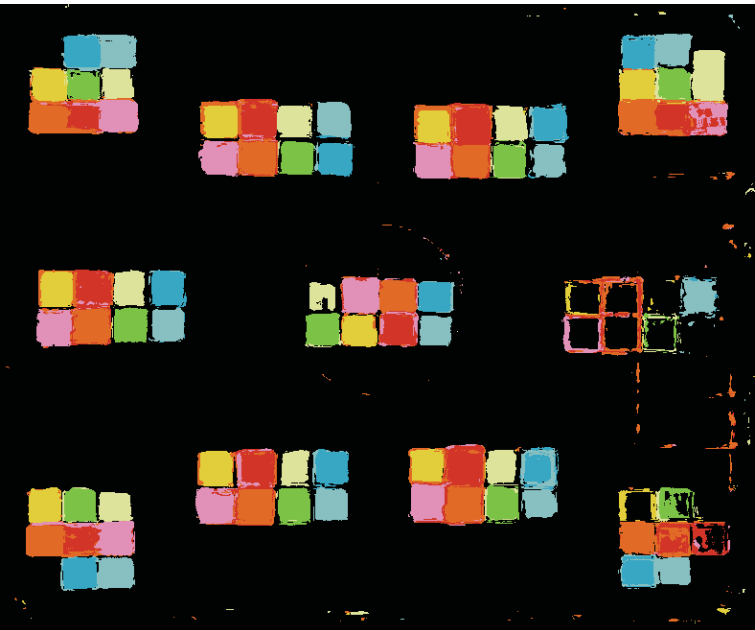
Scenario	Background	Data Set	TP	FP
Uneven Slight	Background Included	Training	98.61%	2.35%
				

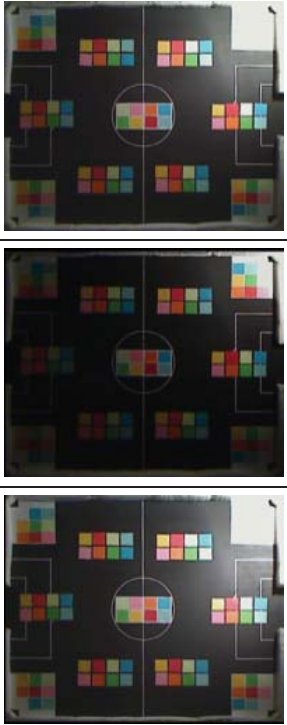
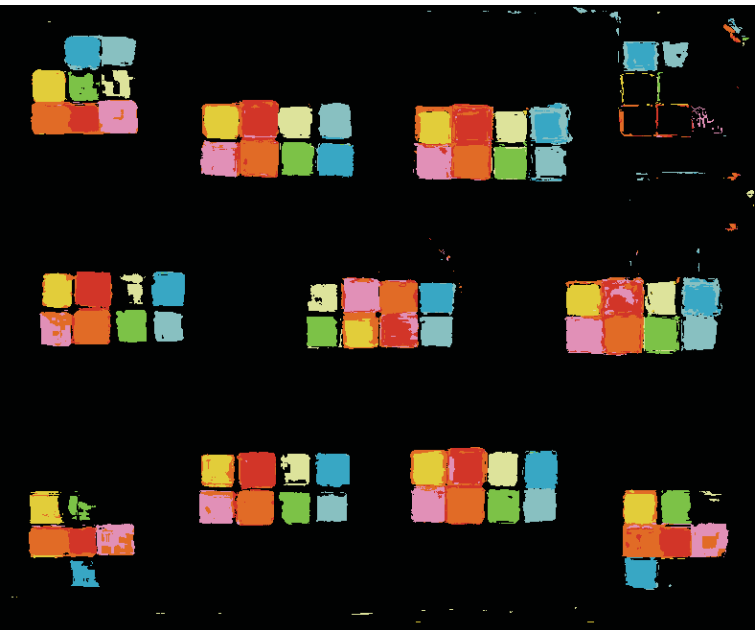
Scenario	Background	Data Set	TP	FP
Even Bright	Background Included	Test	99.38%	3.66%
 				


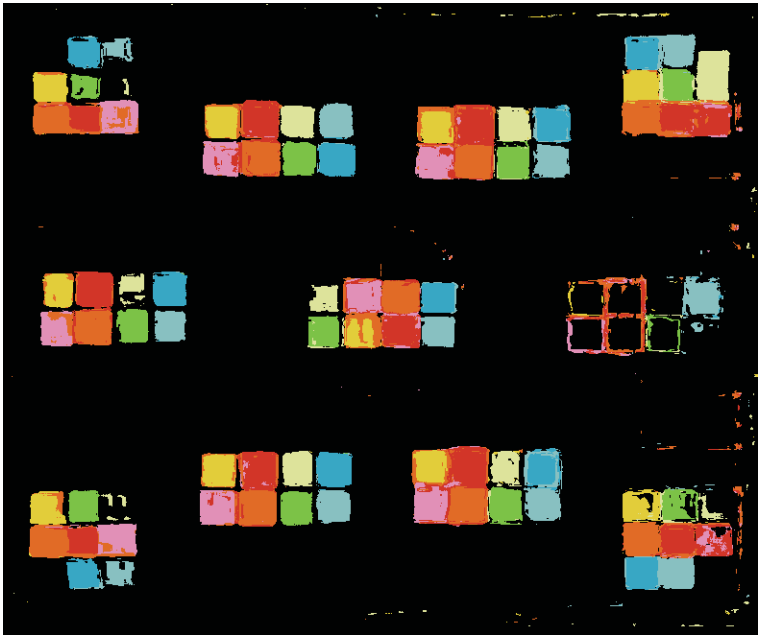
Scenario	Background	Data Set	TP	FP
Even Dim	Background Included	Test	99.51%	0.72%
 				

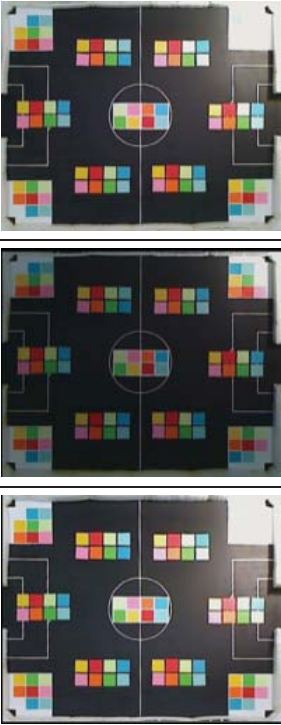
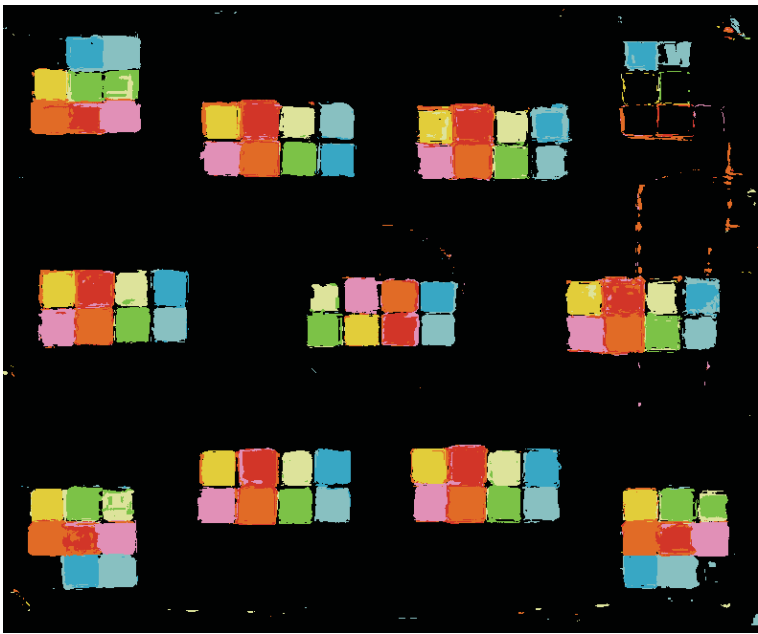
Scenario	Background	Data Set	TP	FP
Even Medium	Background Included	Test	98.56%	2.47%
				

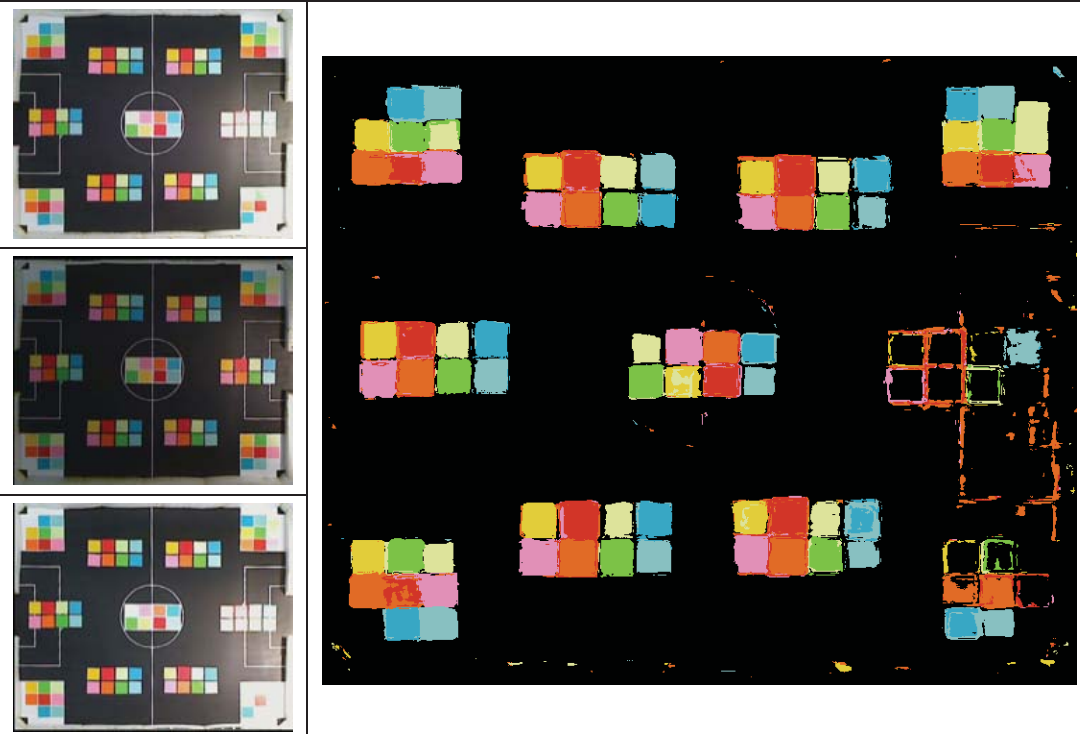
Scenario	Background	Data Set	TP	FP
Harsh Moderate 1	Background Included	Test	89.98%	4.06%
				

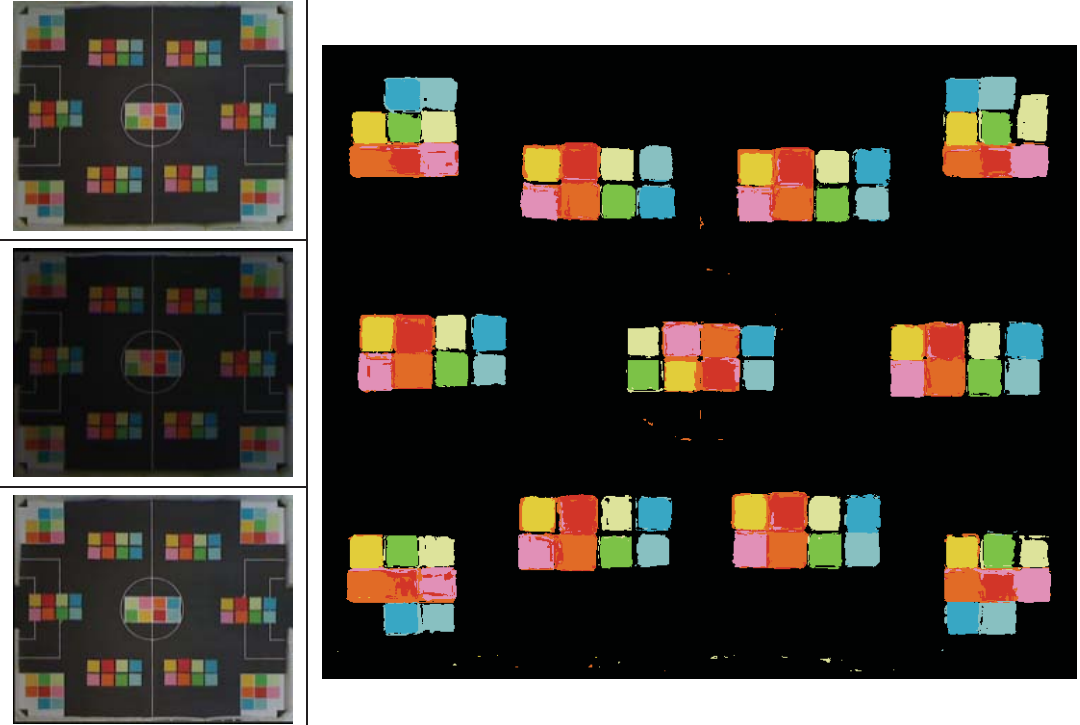
Scenario	Background	Data Set	TP	FP
Harsh Moderate 2	Background Included	Test	85.73%	6.28%
 				

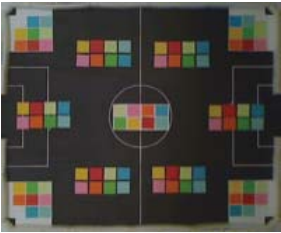
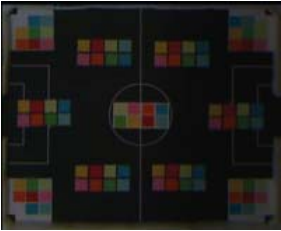


Scenario	Background	Data Set	TP	FP
Harsh Severe 1	Background Included	Test	83.57%	5.18%
 				


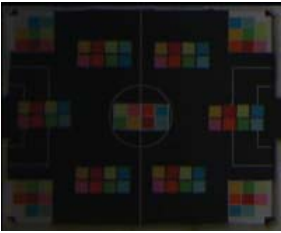

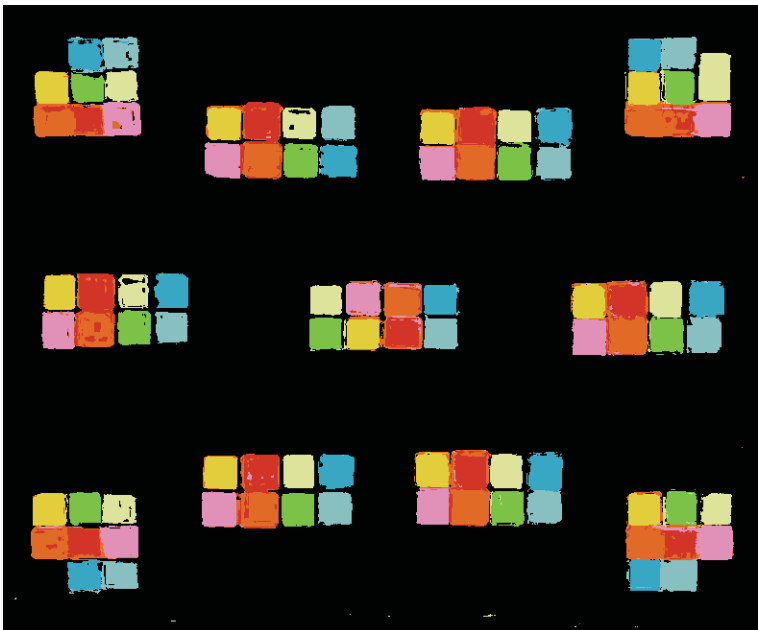
Scenario	Background	Data Set	TP	FP
Harsh Severe 2	Background Included	Test	83.42%	7.73%
 				

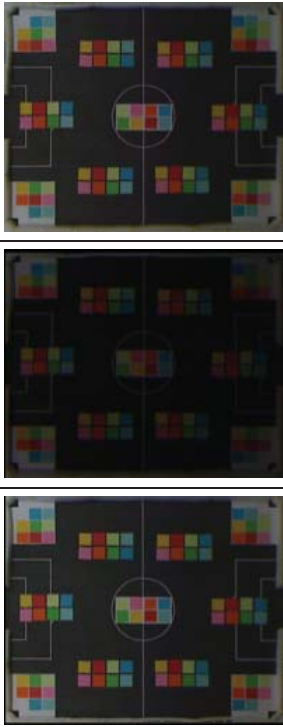
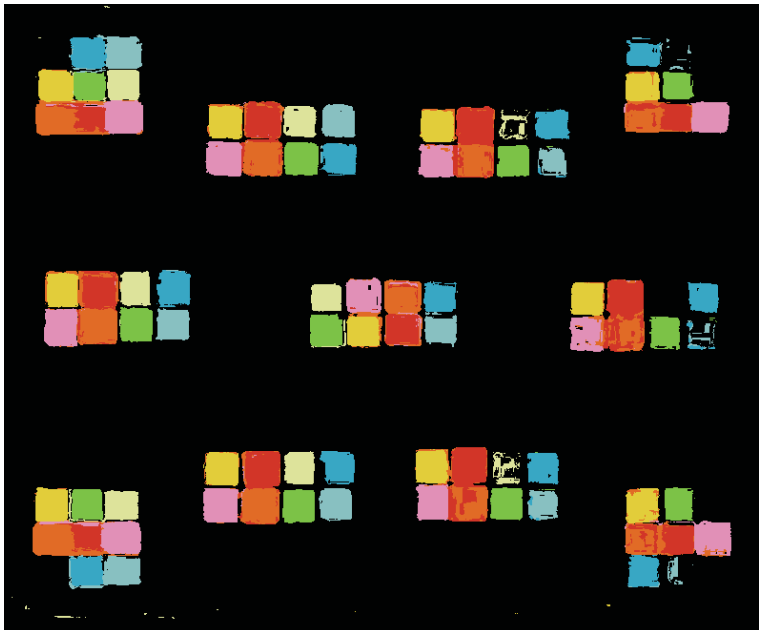
Scenario	Background	Data Set	TP	FP
Harsh Slight 1	Background Included	Test	89.32%	7.01%
 				

Scenario	Background	Data Set	TP	FP
Harsh Slight 2	Background Included	Test	83.42%	11.26%
				

Scenario	Background	Data Set	TP	FP
Uneven Moderate 1	Background Included	Test	98.45%	2.04%
				

Scenario	Background	Data Set	TP	FP
Uneven Moderate 2	Background Included	Test	98.34%	2.34%
  				

Scenario	Background	Data Set	TP	FP
Uneven Severe 1	Background Included	Test	97.92%	1.94%
  				

Scenario	Background	Data Set	TP	FP
Uneven Severe 2	Background Included	Test	90.07%	1.92%
				

Scenario	Background	Data Set	TP	FP
Uneven Slight	Background Included	Test	98.48%	2.85%
