Вся процедура анализа изображения состоит в сравнении каждого пиксела изображения с какими-либо пороговыми значениями.

Для решения задачи была выбрана технология Windows Forms с языком C#. Стандартный подход роботы с изображениями для этой технологии – работа с RGB (**r**ed, **g**reen, **b**lue — красный, зелёный, синий) данными каждого пиксела изображения. Идея состояла в определении ключевых цветов, на основе которых будет происходить анализ, добавлении «чувствительности» к каждому из ключевых цветов и собственно, анализе каждого пиксела изображения.

Подобный метод имел результаты, которые можно было анализировать. Проблема состояла в том, что цветовая модель RGB не идеальна, человеческий глаз плохо отличает оттенки с RGB, такой формат больше подходит для машины. Проанализировав альтернативные варианты, была выбрана цветовая схема HSL (цветовая модель, в которой цветовыми координатами являются тон, насыщенность и светлота).

Для цветовой схемы HSL при анализе каждого пиксела необходимо получить пиксел и его RGB данные по координатам (x;y). Получив стандартными средствами цвет в схеме RGB необходимо с помощью формул конвертировать его в цветовую схему HSL. Конвертирование каждого пиксела в HSL, и последующий его анализ занимает слишком много процессорного времени, поэтому для оптимизации работы было решено не заниматься сравнением каждого пиксела изображения, а воспользоваться так называемым методом «сеток». Изображение делится на условную сетку с заданным размером ячеек.

Анализ изображения проходит в рамках границ сетки, каждый пиксел сетки проходит тот же путь, от получения его RGB значений по его координатам, преобразовании RGB в HSL и потом уже сравнении с граничными значениями.

Первым этапом работы приложения является загрузка изображения. Для этого, при запуске открывается первая форма приложения, которая изображена на рисунке 1.

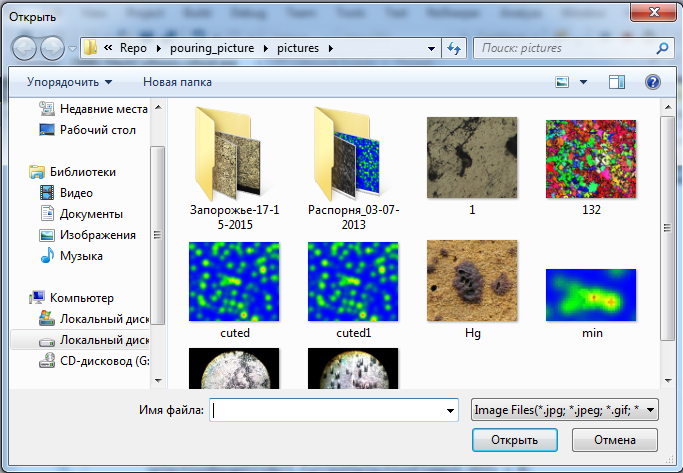


Рис 1. – Форма открытия изображения

После удачной загрузки приложения открывается основная форма приложения, в которой есть возможность подстраивать граничные значения HSL для анализа. Форма изображена на рисунке 2.

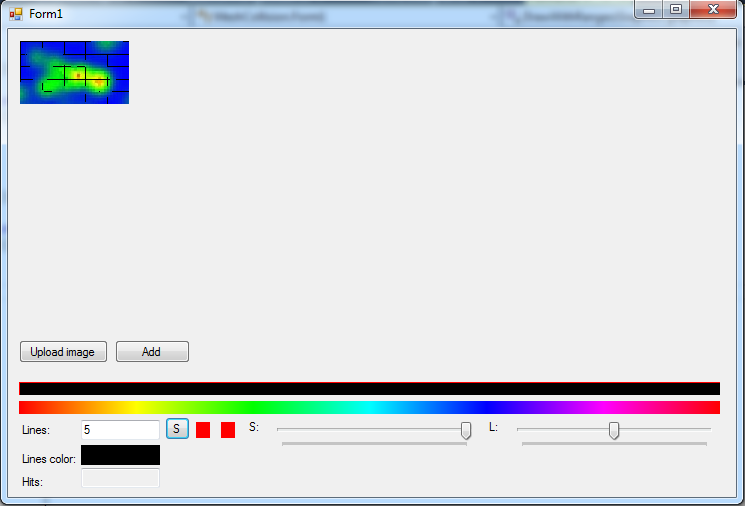


Рис. 2 – Основная форма приложения

Каждая область формы приложения отвечает за свою функциональность. Интерфейс и его функциональность описана на рисунке 3.

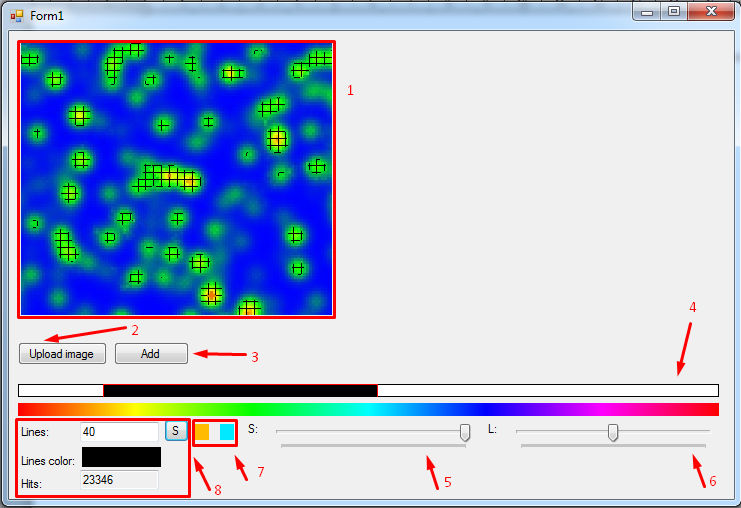


Рис. 3 – Основной интерфейс с описанием

1. Изображение, над которым ведется анализ вместе с сетками, которые были обнаружены.
2. Кнопка, открывающая форму с загрузкой изображения.
3. Кнопка, добавляющая новый регион для анализа.
4. Область, отвечающая за тон искомого региона (H в HSL).
5. Область, отвечающая за насыщенность (S в HSL).
6. Область, отвечающая за светлоту (L в HSL).
7. Область, в которой левая часть отвечает за минимальное значение тона цвета, а правая – за максимальное значение.
8. Область, в которой есть 3 текстовый поля и кнопка. Текстовое поле Lines отвечает за количество линий на каждой из граней сетки (вертикали и горизонтали), Line color отвечает за цвет активного региона сетки, Hits является полем только для чтения, в котором выводиться общая площадь сетки (количество пикселей сетки активного региона). Кнопка «S» отвечает за изменение сетки на основе числа количества линий введенных в текстовом поле «Lines».

Прецедент использования

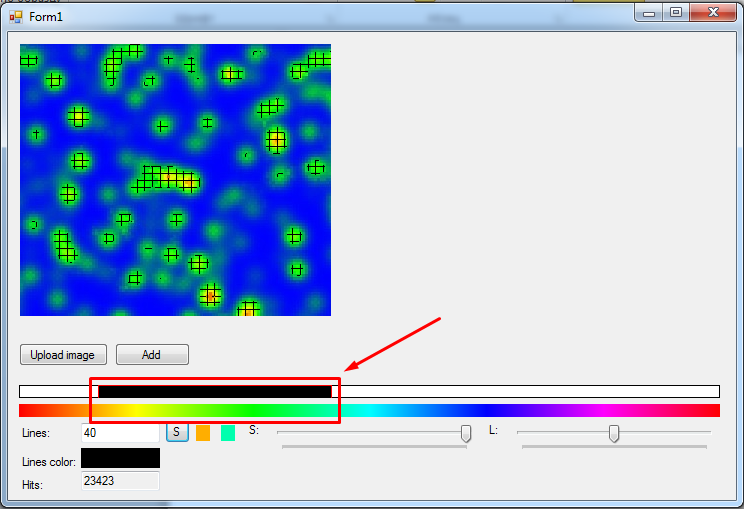


Рис. 4 – Открытая форма с выбранным регионом

1. После открытия основного интерфейса, нажав кнопку «Add» и подкорректируя область выделения получим выделения на всех зелёных регионах. На рисунке 4 изображена форма с выделенным зелёным регионом.

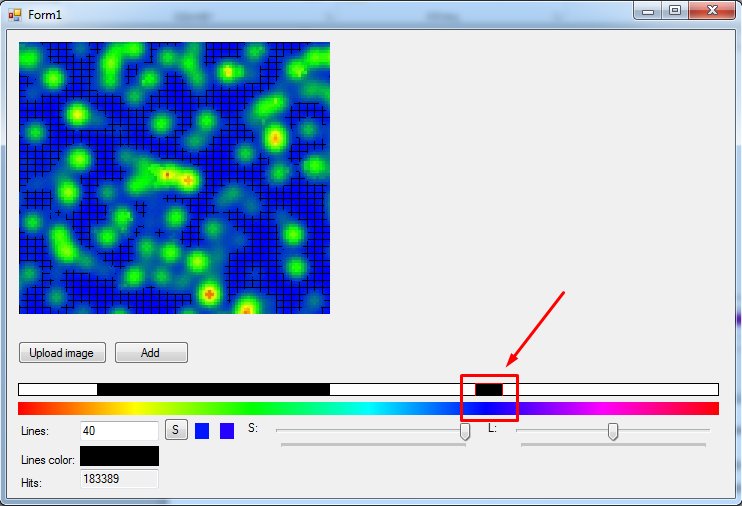


Рис. 4 – Форма с синим выбранным регионом

1. После нажатия кнопки «Add» в область выбора цветового тона добавляется новый регион. Подкорректируя новый регион оставим выделение на синем цвете.

Итог анализа – точки на изображении, которые имеют свои координаты, RGB данные и HSL данные:

1 [X:60 Y:41] [R:192 G:246 B:12] [H:73 S:0,9285715 L:0,5058824]

2 [X:61 Y:41] [R:222 G:253 B:27] [H:68 S:0,9826087 L:0,5490196]

3 [X:62 Y:41] [R:234 G:243 B:28] [H:62 S:0,8995816 L:0,5313725]

4 [X:63 Y:41] [R:239 G:233 B:35] [H:58 S:0,8644068 L:0,5372549]

5 [X:67 Y:41] [R:219 G:230 B:29] [H:63 S:0,8007968 L:0,5078431]

6 [X:68 Y:41] [R:220 G:246 B:23] [H:66 S:0,9253112 L:0,527451]

7 [X:69 Y:41] [R:213 G:253 B:17] [H:70 S:0,9833333 L:0,5294118]

8 [X:70 Y:41] [R:203 G:255 B:13] [H:72 S:1 L:0,5254902]

9 [X:71 Y:41] [R:198 G:255 B:18] [H:74 S:1 L:0,5352941]

10 [X:79 Y:49] [R:197 G:251 B:9] [H:73 S:0,968 L:0,509804]

11 [X:80 Y:49] [R:222 G:255 B:21] [H:68 S:1 L:0,5411765]

12 [X:81 Y:49] [R:225 G:240 B:15] [H:64 S:0,8823529 L:0,5]

13 [X:88 Y:49] [R:231 G:205 B:4] [H:53 S:0,9659575 L:0,4607843]

14 [X:89 Y:49] [R:215 G:218 B:5] [H:60 S:0,9551569 L:0,4372549]

15 [X:90 Y:49] [R:194 G:232 B:9] [H:70 S:0,9253112 L:0,472549]

Сами сетки строятся на основе введенного пользователем количества линий на каждой из граней сетки.

Ввиду того, что изображения в большинстве своём не являются квадратными для вычисления размера ячейки сетки необходимо провести дополнительные вычисления.

weightIndent = Bitmap.Width / linesCount;

heightIndent = Bitmap.Height / linesCount;

,где Bitmap.Width – ширина изображения,

Bitmap.Height – высота изображения,

linesCount – количество линий на каждой из граней, введенное пользователем.

weightIndent и heightIndent – ширина и длина ячейки сетки соответственно.

Получив ширину и длину ячейки, приложение строит сетку по всему изображению.

Для сравнения цветов каждого писеля используются 2 метода. Это либо простое сравнение всех показателей цвета (1), либо сравнение цвета с допуском (2). Выбор применимого метода делается эмпирически, в зависимости от полученных результатов.

return c1.B == c2.B && c1.G == c2.G && c1.R == c2.R;

,где C1.R, c1.G, c1.B – 1й цвет и RGB данные соответственно,

C2.R, c2.G, c2.B – 2й цвет и RGB данные соответственно

1. Метод поиска одинаковых цветов без допуска

if (sens >= 255)

return true;

if (c1.R >= c2.R - sens && c1.R <= c2.R + sens)

if (c1.G >= c2.G - sens && c1.G <= c2.G + sens)

if (c1.B >= c2.B - sens && c1.B <= c2.B + sens)

return true;

return false;

,где sens – допуск цвета,

C1.R, c1.G, c1.B – 1й цвет и RGB данные соответственно,

C2.R, c2.G, c2.B – 2й цвет и RGB данные соответственно

(2) Метод поиска одинаковых цветов с допуском