Image Similarity Search

By Alejandro Gonzalez
Harry Hernandez
Jariel Laureano
Luis Perez

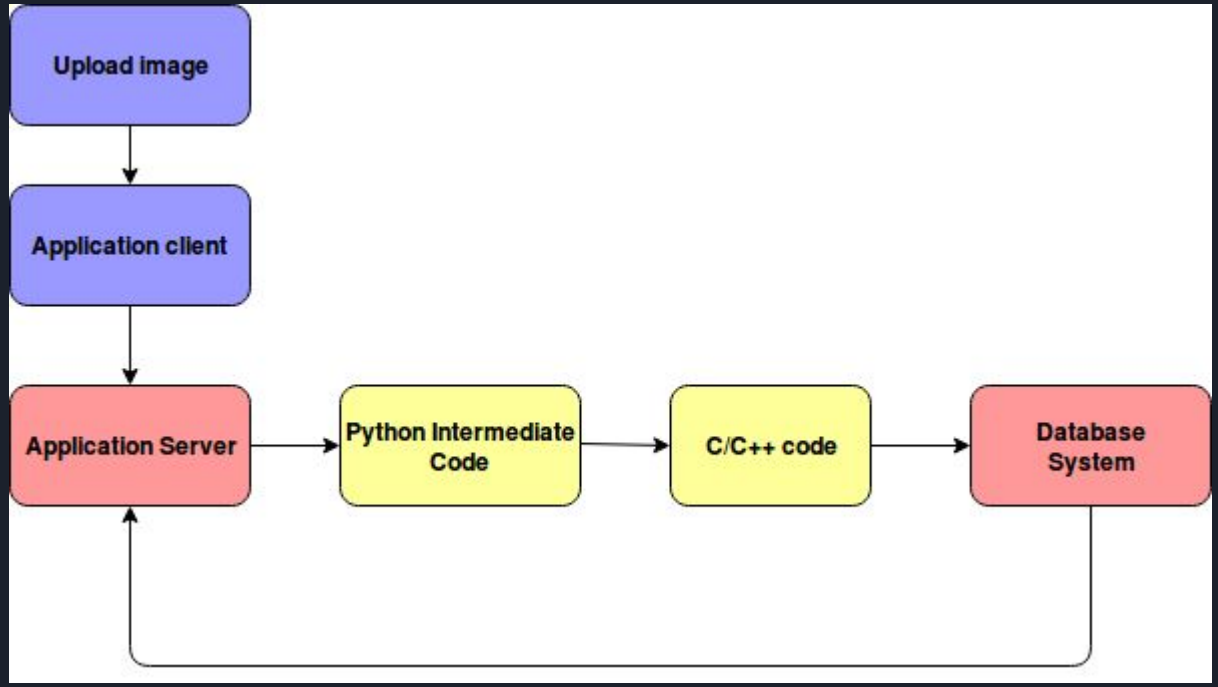
Objectives

- Project Motivation
- Approach
- Implementation
- Experimental setting
- Results
- Conclusions

Project Motivation

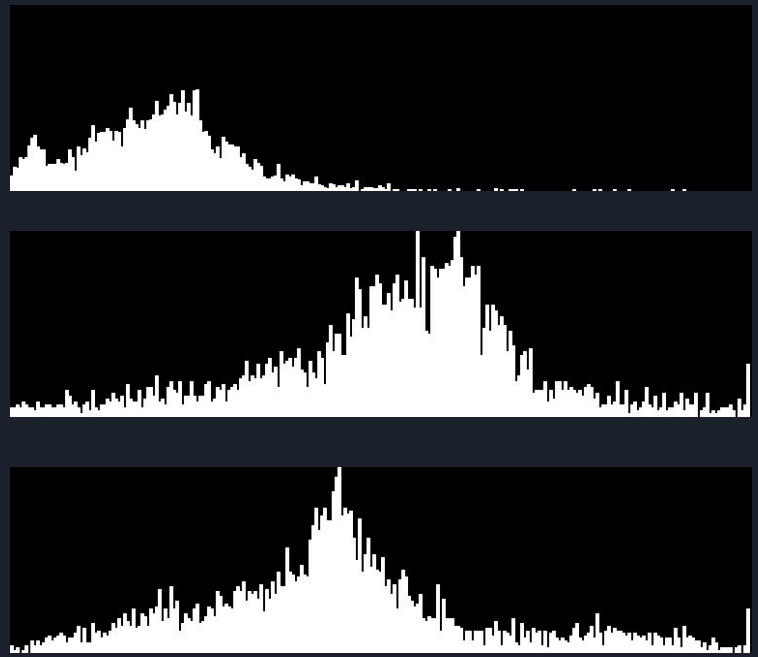
As technology is growing and evolving, engineers are trying to replicate the behavior of the human body. What computers vision does from the perspective of engineering is to seek the automation of tasks that the human visual system does. In image processing, similarity search is a mayor application for computer vision and it includes methods for computing abstractions of an image and making decisions on whether features on different images are similar. Analyzing and implementing one of the algorithms with a high performance computing approach is what we are looking forward to do.

Approach


Upload image
Application client
Application Server
Python Intermediate Code
C/C++ code
Database System

Implementation

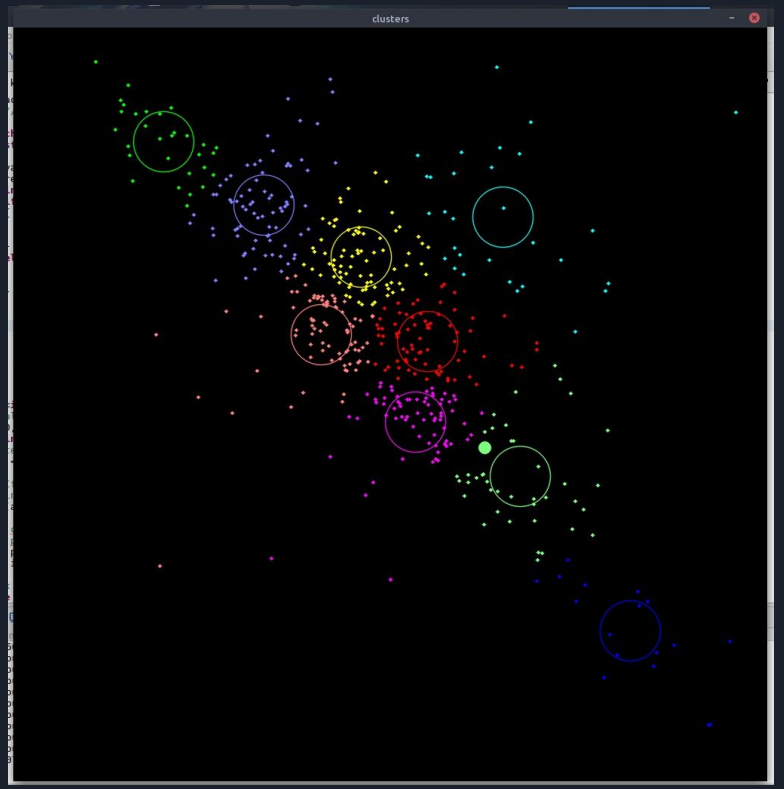
- For image similarity we use Histogram method
  -acts as a graphical representation of the tonal distribution in a digital image.
  -It plots the number of pixels for each tonal value.
- OpenCV-open source computer vision and machine learning software library
- ImageNet -image database organized according to the WordNet hierarchy



Implementation

- K-means

  -Unsupervised learning algorithms that solve the well known clustering problem



Experimental settings

1. Analyze and understand the algorithm in order to simplify parallel algorithm.

2. Create parallel algorithm(OpenMP).

3. Verify if calculations of parallel algorithm equals sequential algorithm.

4. Analyze execution time of both algorithms.

```
double sum;
int size;
size =1048576;
sum = 0;
skew = 0;
variance = 0;
kurtosis=0;
int i;

for(i = 0; i < size; i++){
  sum += histogram[i]*i;
    }

mean = sum / size;


variance = sqrt(mean);

sum=0;

for(i = 0; i < size; i++){
    sum += (pow((histogram[i]-mean)/variance,3)*i);
}

skew= sum/size;


sum=0;
for(i = 0; i < size; i++){
        sum += pow((histogram[i]-mean)/variance,4)*i;
        }

        //-3 for normal dist


kurtosis = ((sum / size)-3);
```

Experimental settings

```
double sum;
int size;
size =1048576;
sum = 0;
skew = 0;
variance = 0;
kurtosis=0;
int i;

for(i = 0; i < size; i++){
  sum += histogram[i]*i;
    }

mean = sum / size;


variance = sqrt(mean);

sum=0;

for(i = 0; i < size; i++){
    sum += (pow((histogram[i]-mean)/variance,3)*i);
}

skew= sum/size;


sum=0;
for(i = 0; i < size; i++){
        sum += pow((histogram[i]-mean)/variance,4)*i;
        }

        //-3 for normal dist


kurtosis = ((sum / size)-3);
```

```
double sum;
int size;
size =4096;
sum = 0;
pskew = 0;
pvariance = 0;
pkurtosis=0;

int i;

#pragma omp parallel for shared(histogram) private(i) reduction(+:sum)
for(i = 0; i < size; i++){
  sum += histogram[i]*i;
    }

pmean = sum / size;


sum=0;
pvariance = sqrt(pmean);

#pragma omp parallel for shared(histogram) private(i) reduction(+:sum)
for(i = 0; i < size; i++){
    sum += (pow((histogram[i]-pmean)/pvariance,3)*i);
}

pskew= sum/size;


sum=0;
#pragma omp parallel for shared(histogram,mean,variance) private(i) reduction(+:sum)
for(i = 0; i < size; i++){
        sum += pow((histogram[i]-pmean)/pvariance,4)*i;
        }


pkurtosis = ((sum / size)-3); //-3 for normal dist
```

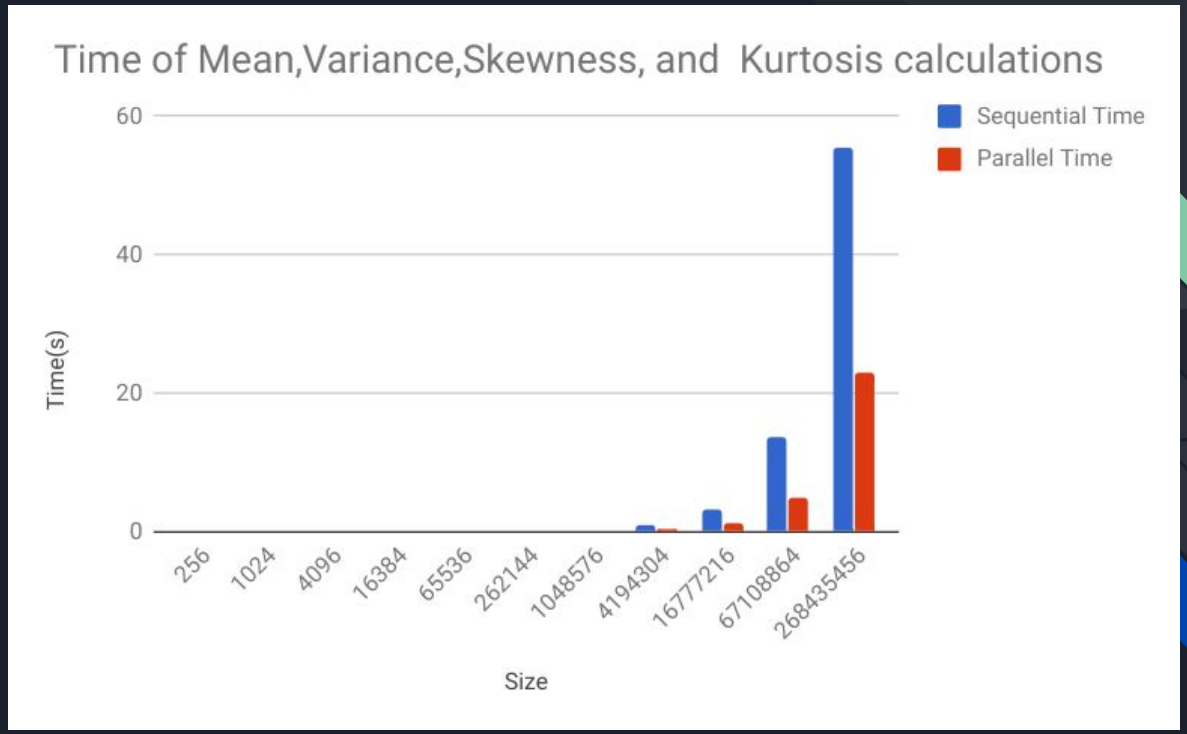
Experimental settings

Verify if calculations are correct

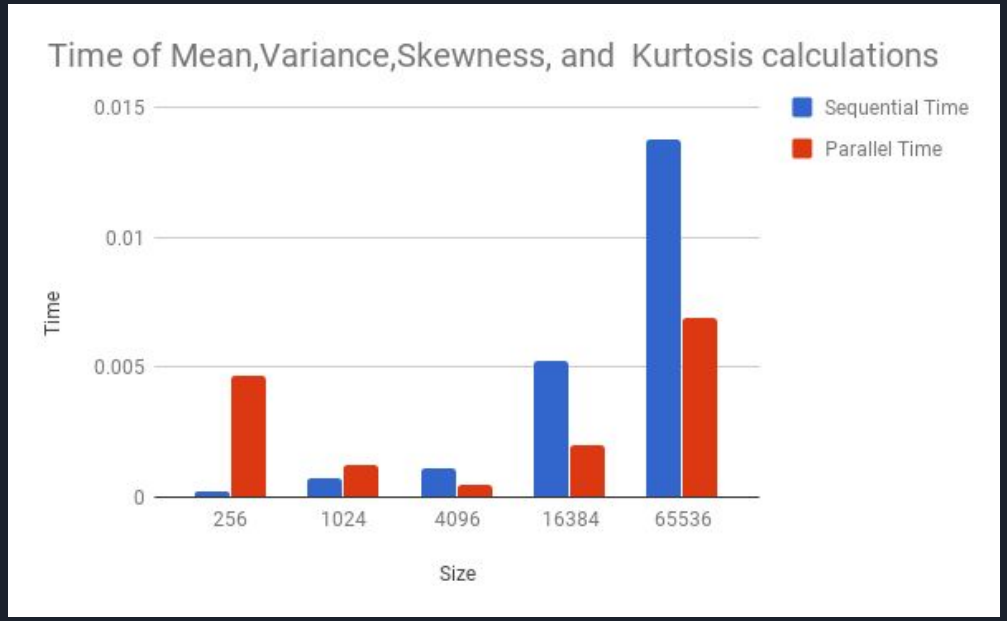
Size = 4096

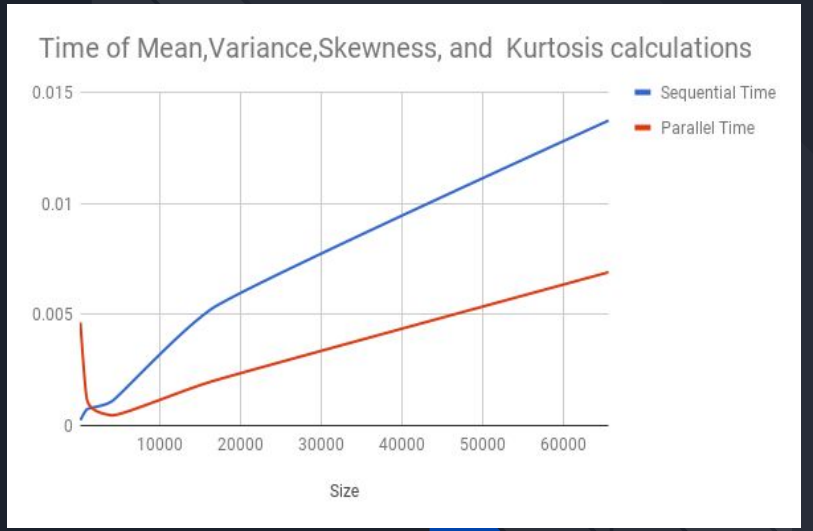
```
          Sequential Calculations ||     Parallel Calculations
          ========================================================
Mean              20935.64                        20935.64
Variance          144.69                          144.69
Skewness          -6193227551.50                  -6193227551.50
Kurtosis          895670264140.07                 895670264140.07
```

Results

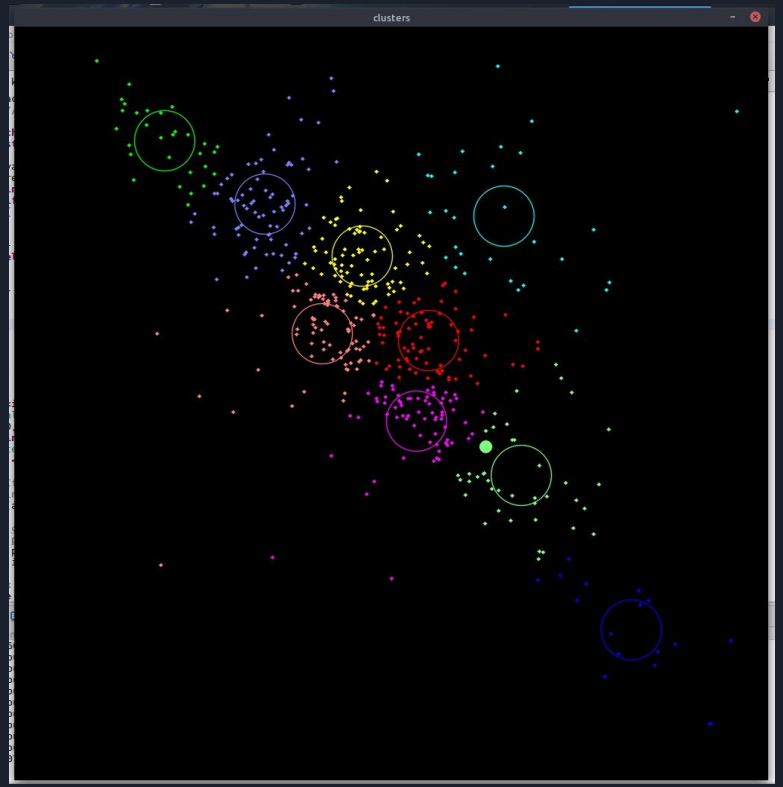

Time of Mean,Variance,Skewness, and Kurtosis calculations
60
40
20
0
Time(s)
Sequential Time
Parallel Time
256
1024
4096
16384
65536
262144
1048576
4194304
16777216
67108864
268435456
Size

Results


Time of Mean,Variance,Skewness, and Kurtosis calculations
Sequential Time
Parallel Time
0.015
0.01
0.005
0
Time
256
1024
4096
16384
65536
Size


Time of Mean,Variance,Skewness, and Kurtosis calculations
Sequential Time
Parallel Time
0.015
0.01
0.005
0
10000
20000
30000
40000
50000
60000
Size

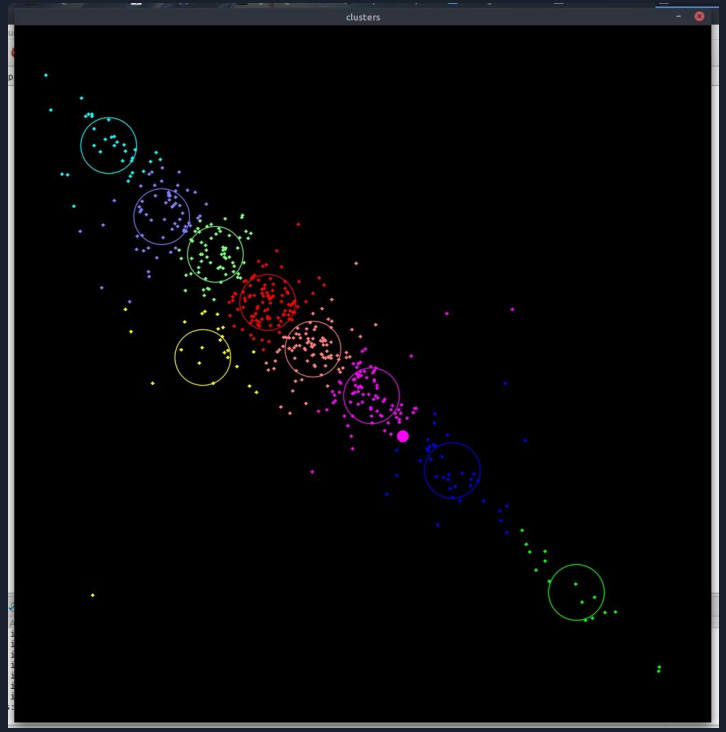
Results

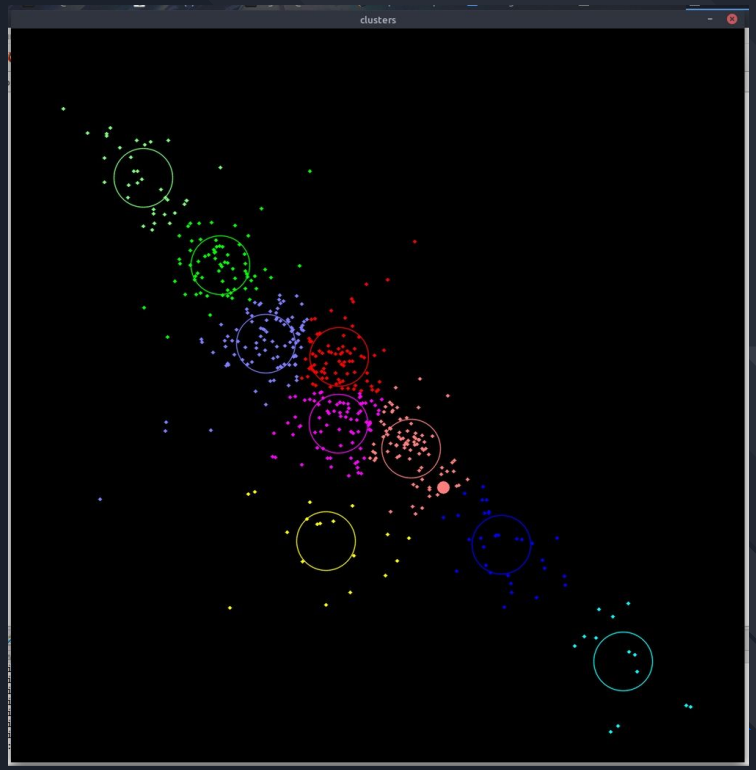



```
GR Cluster id 0, count 80
GR Cluster id 1, count 67
GR Cluster id 2, count 31
GR Cluster id 3, count 96
GR Cluster id 4, count 84
GR Cluster id 5, count 34
GR Cluster id 6, count 13
GR Cluster id 7, count 78
GR Cluster id 8, count 18
Compactness: 1.6131e+06
BG Cluster id 0, count 87
BG Cluster id 1, count 77
BG Cluster id 2, count 36
BG Cluster id 3, count 84
BG Cluster id 4, count 23
BG Cluster id 5, count 25
BG Cluster id 6, count 52
BG Cluster id 7, count 101
BG Cluster id 8, count 16
Compactness: 1.45941e+06
RB Cluster id 0, count 31
RB Cluster id 1, count 76
RB Cluster id 2, count 76
RB Cluster id 3, count 12
RB Cluster id 4, count 28
RB Cluster id 5, count 104
RB Cluster id 6, count 84
RB Cluster id 7, count 27
RB Cluster id 8, count 63
Compactness: 2.67777e+06
del ok, key: simple_room-wallpaper-4096x3072.jpg, ret: 1

Common 0 pos 0 name simple_room-wallpaper-4096x3072.jpg
Common 1 pos 5 name n03970156_96.JPEG
Common 2 pos 6 name n03970156_95.JPEG
Common 3 pos 13 name n03970156_89.JPEG
Common 4 pos 26 name n03970156_77.JPEG
Common 5 pos 31 name n03970156_72.JPEG
Common 6 pos 34 name n03970156_6.JPEG
Common 7 pos 49 name n03970156_56.JPEG
Common 8 pos 53 name n03970156_52.JPEG
Common 9 pos 59 name n03970156_498.JPEG
Common 10 pos 87 name n03970156_472.JPEG
Common 11 pos 93 name n03970156_467.JPEG
Common 12 pos 101 name n03970156_45.JPEG
Common 13 pos 107 name n03970156_454.JPEG
Common 14 pos 116 name n03970156_446.JPEG
Common 15 pos 147 name n03970156_418.JPEG
Common 16 pos 161 name n03970156_405.JPEG
Common 17 pos 164 name n03970156_402.JPEG
Common 18 pos 206 name n03970156_365.JPEG
```

Results





Conclusion

We achieved to find images similitude, using features like colors, contrast, and brilliance. We utilize a small image net , we used 9 clusters on 3 dimensions.