CSC 249 Final Project Report

Chris Dalke & Nate Conroy

May 15, 2017

1 Introduction

For our final project in CSC 249, we analyzed the correlation between the weather of the top 50 largest cities in the United States and the emotions of its occupants as seen from geotagged photos taken within those cities.

This two-part experiment first involved retrieving a weather dataset for the cities that we chose to analyze. We used public weather databases to fill in a database of weather conditions for the top 50 US cities.

The second part of the experiment involved retrieving a specified amount of publicly available photos geotagged within each of the cities that contain faces. We used the Flickr API to retrieve these photos. We then analyzed the faces to determine the mood of the subject and accumulate the results to calculate average emotion scores for each of the cities. Lastly, we charted the average emotion scores for each location against the weather scores, to obtain a comparison between the two variables and see if they are correlated.

1.1 Running our System

For convenience, all relevant analysis from the visualization system has been placed into this document. We have also included a large number of screenshots in the `Screenshots` folder.

Our system consists of two pieces: A set of Java programs which performed the data collection, and an HTML/JS visualization system.

The Java program is contained in the Code folder, but it is unnecessary to run this program since the results have already been saved into a JSON file.

The HTML/JS visualization system can be run, if desired, by running an HTTP server (For example the node.js `http-server`) on the Visualization folder, and opening index.htm via your web browser. *Note: Opening index.htm directly will not load the data, because the JS must have access to the files in the surrounding folder, which requires a web server to be running on the local machine.*

To make it easier to use the visualization system, I have hosted the tool at `http://chrisdalke.com/csc249`. That URL will open the full visualization tool with our final dataset and customizable chart tools.

2 Procedure

2.1 Data Collection

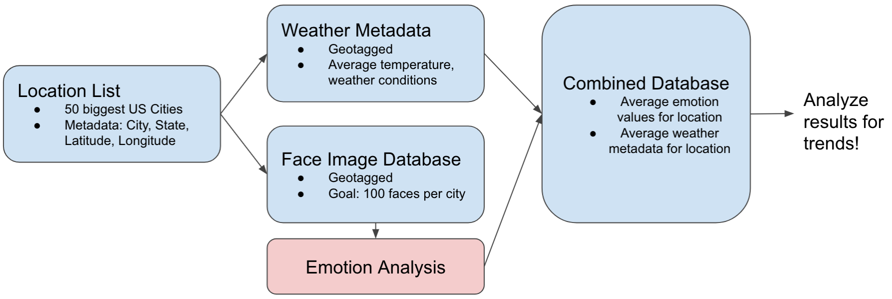

Figure 1: Overview of the Data collection process

2.1.1 Location Dataset

For our location dataset, we compiled a list of the top 50 largest US cities by population as well as their centralized latitude and longitude coordinates. We formatted the data into a CSV file, which we then parsed and used to create a JSON file filled with locations and their metadata.

When deciding which set of locations we would analyze, we decided that rural areas are unlikely to have a significant amount of geotagged photos, making data collection difficult. Instead, we would only look at cities because these locations are most likely to have a large amount of data.

2.1.2 Weather Dataset

Our second dataset was built on top of the location dataset, and added weather metadata to each city so that we could analyze weather conditions against facial emotions.

We chose not to analyze cityscape images to determine weather, because localized weather conditions are not indicative of overall weather conditions, so analyzing weather in photos would not give us meaningful long-term weather data.

Instead, we used free online weather and climate databases to fill in statistics for each city. Our primary source of aggregated climate data was *USClimate-Data.com*. We collected the following weather statistics for each location:

Field Name	**Description**
hoursSunshine	Hours of sunshine the location receives per year.
rainfall	Inches of rainfall per year.
snowfall	Inches of snowfall per year.
averageTempHigh	Average daily high temperature over a year.
averageTempMid	Average daily median temperature over a year.
averageTempLow	Average daily low temperature over a year.
daysRain	Days on which it rains in a year.

2.1.3 Face Photo Dataset

We obtained our dataset of photos using the Flickr API, which allowed us to create a search query using the latitude and longitude coordinates we compiled and stored in the locations JSON file. We decided to utilize the sort feature 'Interestingness', rather than the default chronological order, in our query to avoid getting an disproportionate amount of data from one user. We were also able to set an encompassing radius around the centralized city location, that would return more or less photos depending on our input. Lastly, we used the tag "person" in our query to have an initial filter on our raw dataset of images which would help remove photos that didn't contain a person.

Our data collection program queries the Flickr API for new photos and analyzes the faces in each photo until the database reaches a minimum size of 100 faces for each city.

A recommendation that we were given by Professor Luo was to expand our dataset to 50k images instead of 5000, but we were unable to do that due to limitations described in *3.1.3 - Data Set Size*.

2.2 Emotion Calculation

Once we built our face photo dataset, we needed to calculate the emotion values for each face in each image, store those values, and average them together to get an emotion value for each city.

We chose to use the Microsoft Emotion API, which allows a user to pass in a URL to an image, and the API analyses all faces it finds in an image for emotions. The Emotion API is easy to use and fits within our time constraints.

Sample output from the Emotion API is shown below. The API recognizes 8 emotions:

- Anger
- Contempt
- Disgust
- Fear
- Happiness
- Neutral
- Sadness
- Surprise

For each face, the API returns an array of decimal probabilities representing the confidence the algorithm has that the given face is each emotion. An example output can be seen below.

```
{
    "faceRectangle": {
      "left": 488,
      "top": 263,
      "width": 148,
      "height": 148
    },
    "scores": {
      "anger": 9.075572e-13,
      "contempt": 7.048959e-9,
      "disgust": 1.02152783e-11,
      "fear": 1.778957e-14,
      "happiness": 0.9999999,
      "neutral": 1.31694478e-7,
      "sadness": 6.04054263e-12,
      "surprise": 3.92249462e-11
    }
  }
```

Figure 2: An example of the Emotion API output in JSON format

Once we calculated the emotion for each individual image, we aggregated the emotion values for each city by adding them together and normalizing them. As a result of this process, each city has an emotion matrix representing the average emotions of the photos collected for that city.

2.3 Data Cleanup

Once we collected the photo dataset, we trimmed out all the photos that did not contain faces; our database contained about a 1:20 ratio of face to non-face images, so we trimmed out the non-face images so the visualization tool would only load face images. Additionally, we built JSON files containing all of the additional emotion and weather metadata for each location in a single file.

2.4 Data Visualization

We needed a way to analyze the data we produced and display it in a convenient manner that would allow us to produce charts for analysis. The main objectives of this system are to ensure the program is identifying all faces and correct emotions in the images, and to dynamically generate charts that could be used to visualize and analyze the data. We built a visualizer which loads the JSON databases, allows browsing of the dataset, and displays customizable charts which can be exported and are used in this final report.

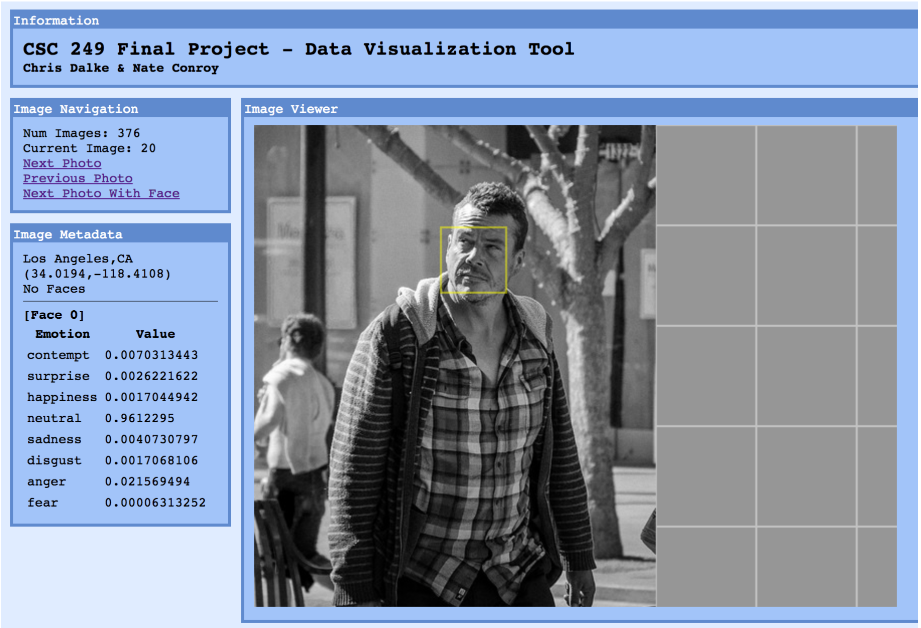

Figure 3: Screenshot of our data visualization tool displaying emotion values for a face

Figure 4: Chart functionality of our data visualization tool

2.5 Software Utilized

We used numerous technologies to build our program. To perform data collection and conversion, we used a Java program. The Java program parsed the raw CSV input for the location and weather datasets and used the Flickr API and Microsoft Emotion API to build the facial image dataset.

To visualize our results, we used a program built with HTML/JS that loads and displays the JSON data. We used a JS charting library, Highcharts, which helped to create custom bar charts and scatter plots, as well as calculating the line of best fit for each type of data.

3 Results & Analysis

3.1 Data Sets

Our final face & emotion data set contains 3858 images, with 4849 faces. Our final dataset was 96% of our goal of 5000 faces. The dataset also includes aggregated emotion values for each city, as displayed in the figure below.

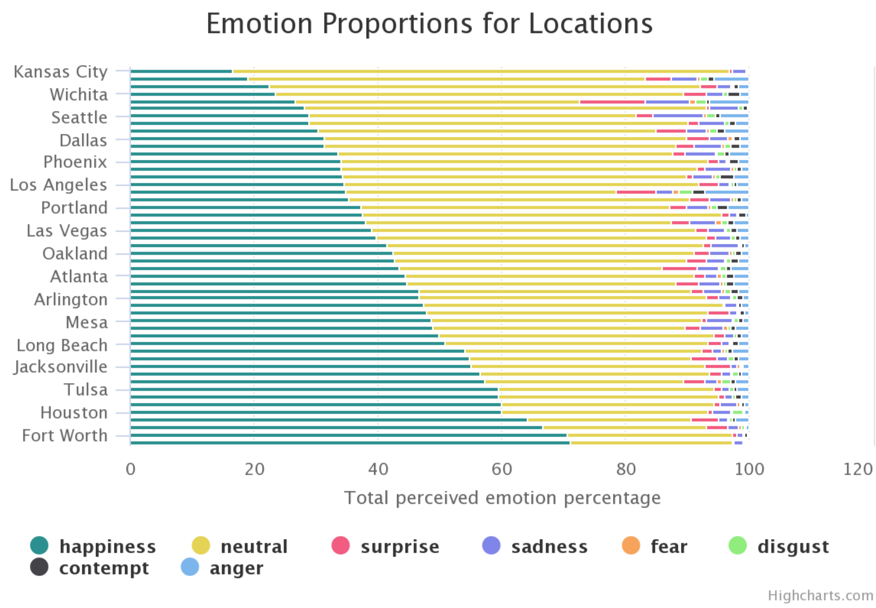

Figure 5: Aggregated emotion proportions for cities (Some labels omitted)

The happiest city was Nashville, Texas, with a happiness value of 0.711. The least happy city was Kansas City, Missouri with a happiness value of 0.166.

3.1.1 Success Cases

We were particularly satisfied with the quality of our data set: The system accurately identified emotion values in a large proportion of the images. A manual skim of the data also led us to conclude that the emotions identified in the faces were very close to what a human would perceive. Examples of the system output are shown below.

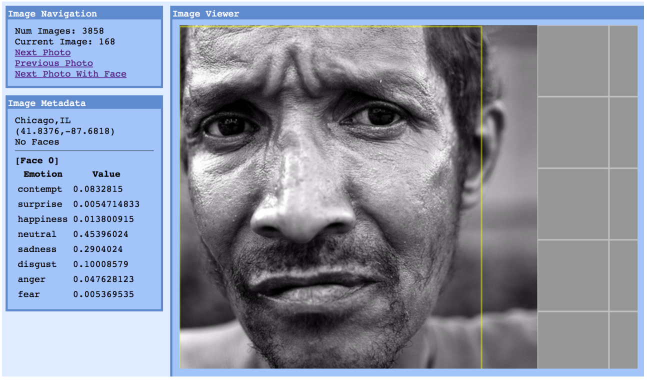

Figure 6: Example of a success case

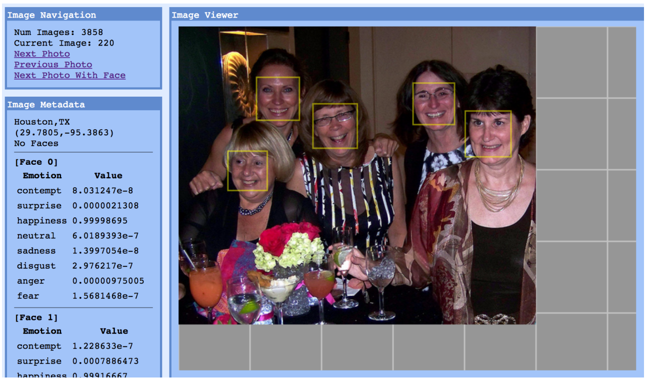

Figure 7: Example of a success case with a photo containing multiple faces

3.1.2 Failure Cases

There were three main failure cases:

1. **The system can not recognize a face**, this was not a problem because it could just find a different face image to reach the total. The image simply was not included in the dataset.

2. **The system misidentifies an emotion**, this problem seemed to occur equally on all locations, so while it does bias the overall emotion proportions, it should not affect the results significantly. Images in which people made silly faces were the main contributing factor for this failure case.

3. **The system identifies faces on non-human objects**, such as statues and billboards. Similarly to the second case, this occurs equally on all locations and is hard to filter out automatically.

Each of these cases occurred at a low rate (less than 1%), so we decided it was not worth manually filtering them. Examples of the failure cases are shown below.

Figure 8: Example of the system failing to recognize a face (right)

Figure 9: Example of the system misidentifying an emotion

Figure 10: Example of the system falsely identifying a human face

3.1.3 Data Set Size

If we wanted to build a more accurate analysis, using more photos would be beneficial. Our first obstacle to getting more images is time - The Emotion API's free Tier rate limits requests to 30k a month, which we hit during the course of finding face images. We found that there was a ratio of 1:20 face images to non-face images, so finding 5000 faces images required 100k images, which hit the rate limiting on all the developer accounts our group was able to create. Additionally, Flickr rate limits geo queries. These two factors made it difficult to obtain a larger dataset given the time constraints.

Professor Luo had recommended approaches to filtering our dataset to avoid "tourist" photos and expanding our radius. We were able to expand the radius for cities that could not find very many photos and we also defined the queries to search over a longer period of time. However, we were unable to filter out tourist or professionally taken photos in the given time constraints.

3.2 Analysis

The data points can be graphed on a 2D plane, where, given a choice of emotion and weather variable, the X axis is the weather variable and the Y axis is the emotion variable. This produces a scatter plot, which we can then use to analyze the relationship between the variables mathematically.

3.2.1 Regression Analysis

Since the relationship between weather factors and emotion values would likely be linear, for our statistical analysis we chose to fit the data to a linear line of best fit. The results of every combination are shown below, with their R-Squared values.

Weather Variable \ Emotion Variable	Anger	Contempt	Disgust	Fear	Happiness	Neutral	Sadness	Surprise
hoursSunshine	0.19	0	0	0	0.14	0	0	0
rainfall	0	0	0	0	0.1	0	0	0.13
snowfall	0	0	0	0	0.34	0.44	0	0
averageTempHigh	0.42	0	0	0	0.26	0.18	0	0.18
averageTempMid	0.23	0	0	0	0.14	0.06	0	0.09
averageTempLow	0	0	0	0	0.16	0.05	0	0
daysRain	0.3	0	0	0	0.05	0.11	0.32	0.2

Figure 11: Correlations between all Weather and Emotion variables

Some of the charts for the highest correlations are shown below. NOTE: Charts between all the emotion variables and weather variables are available in the Code/Dataset ZIP file, in the `Screenshots` folder.

Cities without snowfall have been omitted from the calculations for correlation with snowfall. Additionally, our charts exhibit visual artifacts due to floating-point inaccuracies in the JS chart library, but the R-Squared value is calculated for a linear line of best fit.

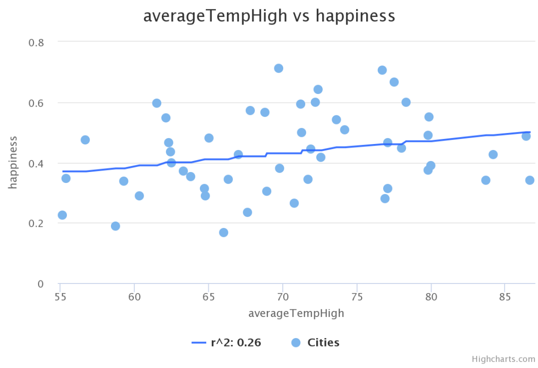

Figure 12: Average Temp High vs. Happiness

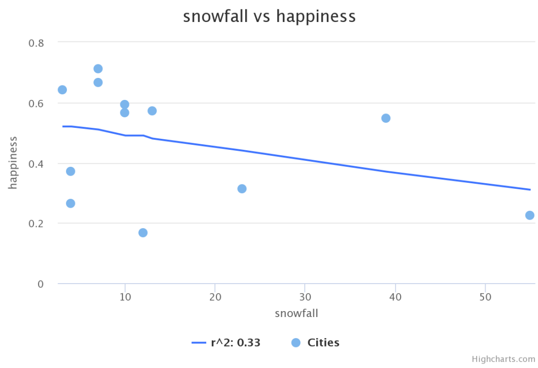

Figure 13: Snowfall vs. Happiness

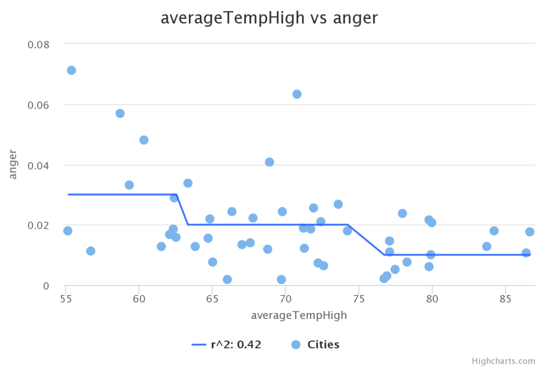

Figure 14: Average Temp High vs. Anger

3.2.2 Downsides of Procedure

Machine and human vision both make judgments from limited data, and the perceived emotion of a person in an image doesn't necessarily correlate with their actual happiness. Additionally, our data set has some disadvantages: Most of the photos we were able to collect online were likely taken by professional photographers who use Flickr to host their photos, meaning that the emotions in their photos are biased The limitation is also on the medium of photography. We still found that analyzing perceived emotion was a valuable machine vision project, since we were exposed to the challenges of building a large data set and performing computer analysis on the images.

3.2.3 Conclusion

While our project delivered interesting results, we were unable to find a statistically significant trend among the resulting the data. The highest R-Squared value was 0.44, which is too low to claim statistical significance between the weather data we collected and emotion of that location. There are many other factors that could affect the person's perceived emotion, including the event they are attending, whether the photo is a candid or not, and chance. We could not isolate weather and geolocation as a driving factor in perceived emotion based on the given statistical evidence.

Further work in increasing our dataset size may prove helpful in demonstrating a correlation between emotion and weather data.

4 Works Cited / Previous Work

Roser, Martin, and Frank Moosmann. "Classification of weather situations on single color images." IEEE Intelligent Vehicles Symposium (2008): 798-803. IEEE Xplore Digital Library. Web. 27 Mar. 2017.

McDonald, Kyle and Saragih, Jason, FaceTracker, (2017), GitHub repository, https://github.com/kylemcdonald/FaceTracker

Suchitra, Suja P. and S. Tripathi, "Real-time emotion recognition from facial images using Raspberry Pi II," 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, 2016, pp. 666-670.

Dhiraj Joshi, Ritendra Datta, Elena Fedorovskaya, Jia Li, James Z. Wang, Jiebo Luo, "Computational inference of aesthetics, mood, and emotion in images," IEEE Signal Processing Magazine, 28(5): 94-115, September 2011.

Hua Wang, Dhiraj Joshi, Jiebo Luo, and Heng Huang, "Simultaneous Image Annotation and Geo-Tag Prediction via Correlation Guided Structured Multi-Task Learning," IEEE Symposium on Multimedia (ISM), December 2012.

5 Academic Honesty

We did not collaborate on this assignment with any other groups and all work is our own.