**ABSTRACT**

In this world, crimes are an inseparable part of our lives. Every day we hear about them and some of us even involved in at least one of them during our life. Being cautious and improve safety is not a simple instruction anymore. We need to use modern technology and data science techniques to more wisely act against this problem. There are so many records and documentation in the police department that have been gathered during the years, which can be used as a valuable source of data for the data analytics tasks. Applying analytical task to these data bring us valuable information that can be used to increase the safety of our society and lower the crime rate.

**TABLE OF CONTENTS**

1. **INTRODUCTION**

**1.1 MOTIVATION**

. The motivation behind this project is to help thePolice Department make better decisions with the use of data.It’s important for them to receive key insights from this dataset. This dataset contains information from the Chicago Police Department from 2012 to 2017.

**1.2 OBJECTIVES**

1. How has the number of various crimes changed over time in Chicago?

2. How have the number arrests corresponded to the crimes changed over time in Chicago?

3. Are there any trends in the crimes being committed?

4. Which crimes are most frequently committed?

5. Which locations are these frequent crimes being committed to?

6. Are there certain high crime locations for certain crimes (etc Sexual offense)?

7. How has the number of certain crimes (etc homicide) changed over the years in Chicago? 8. How do we keep people aware about the dangers in their areas? 9.How can we track them?

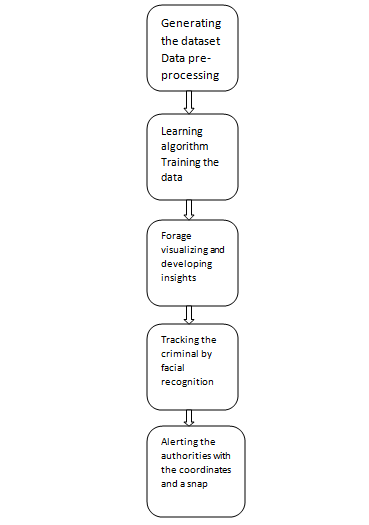
**2. GENERATING THE DATA SET**

**2.1 Data Exploration**Data is extracted from the Chicago Police Department’s CLEAR (Citizen Law Enforcement Analysis and Reporting) system. In order to protect the privacy of crime victims, addresses are shown at the block level only and specific locations are not identified”.In general, the data included information such as date/time when the crime happened, the block where the crime occurred, type of crime, location description, whether there was an arrest, and location coordinates. In continue we bring more specifically the features of our data.The data had 7,939,202 number of records and 24 columns. The list of the names of each column from left to right are as follows ID, Case Number, Date, Block, IUCR, Primary Type, Description, Location Description, Arrest, Domestic, Beat, District, Ward, Community Area, FBI Code, X Coordinate, Y Coordinate, Year, Updated On, Latitude, Longitude, Location

* 1. **Data Extraction**

Chicago Crime dataset requires one of the most important data pre-processing procedure which is cleaning. Our data need to be clean by: 1.Removing duplicate rows2.Removing missing values (etc. Null/NA values) in the dataset3.Filtering out all the features from the dataset that are not relevant to our data analysis (etc. X Coordinate, Y Coordinate, Latitude Longitude).To apply these preprocessing tasks on our dataset we used python pandas is this sequence: First, we had a lot of corrupted data in each record that we had to remove. For instance, out of 1,923,865 records on one file 70,627 records filtered due to not matching with the column attribute. After that, we had to find the wrong data in each column and remove them to have a cleaner data. Then we removed the duplicate data, and finally, we delete the column we did not need so our data would be smaller and faster to work with

1. **PROJECT FLOW**

****

**Data processing**: The carrying out of operations on data, especially by a computer, to retrieve, transform, or classify information ,downloading the data from the following link <https://data.cityofchicago.org/> .The data now undergoes data cleaning removal of empty rows as apart of data preprocessing to obtain the final data.

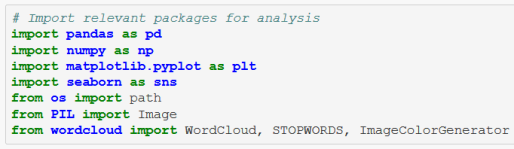
**Learning algorithm Training the data**: the data is then subjected as training data to the machine using naïve bayes classifier for machine learning and testing the data for predictive analysis.

**Forage visualizing and developing in sights**: In this phase we would forage the data explore the data and then visualize the data in th form of graphs and histograms ,devolping insights and action course to be taken based on the result devolping the heat map>

**Tracking the criminal by facial recognition:** we would be taking the photos of the criminals in the various angles and feeding it to the machine to the train the data and we would be using local binary photo histogram through which it would identify the criminal

**Alerting the authorities with the coordinates and a snap**: The mail would be sent to the nearest police station with the snap and the co ordinates of the location where the criminal was spotted .

we will be using python’s framework to help us with the analysis of the data. The different frameworks and packages that we used are shown below.

****

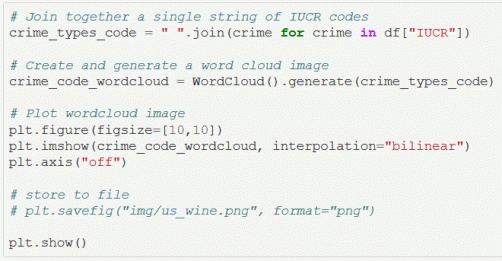
* Pandas: Library used for data manipulation and analysis
* Numpy: Library with high-level mathematical functions
* Matplotlib: Library to plot graphs
* Seaborn: Library built on Matplotlib for visualization.

We extract the data into Pandas dataframe and apply formatting to set the time as hour, month, and year. We have also chosen to exclude 2017’s data points since only Jan 2017’s data is available for the year.

**4.DEVOLPING INSIGHTS**

#### **4.1MOST FREQUENT OCCURING CRIMES**

#### With our dataset, we can group the data together based on Illinois Uniform Crime Reporting (IUCR) code, which is a four digit codes that Chicago’s law enforcement agencies use to classify criminal incidents when taking individual reports. Next, we generate a wordcloud with the grouped data and plot the result.



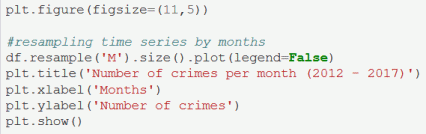
The wordcloud image tells us that “051A” & “031A” are the most frequently occurring IUCR codes in Chicago. We can find what the crime codes represent from the Chicago data link portal.

**051A: Assault - Aggravated Handgun**

**031A: Robbery - Armed Handgun**

**4.2 YEARLY CRIME TREND ANALYSIS**

We are going to analyze the general trend of the crime data from 2012-2016. In order to reduce processing time, we used a resampling method by month for the number of crimes. The resampling method in pandas is similar to the groupby method for a certain time span.



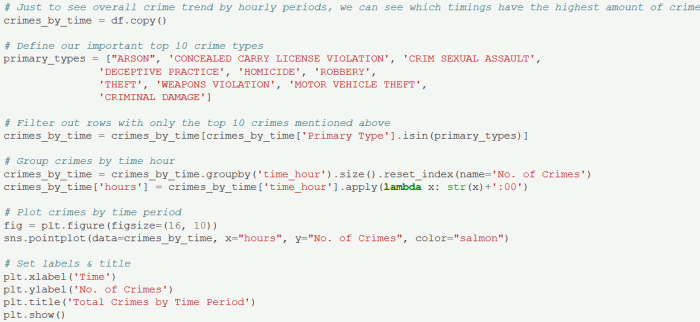


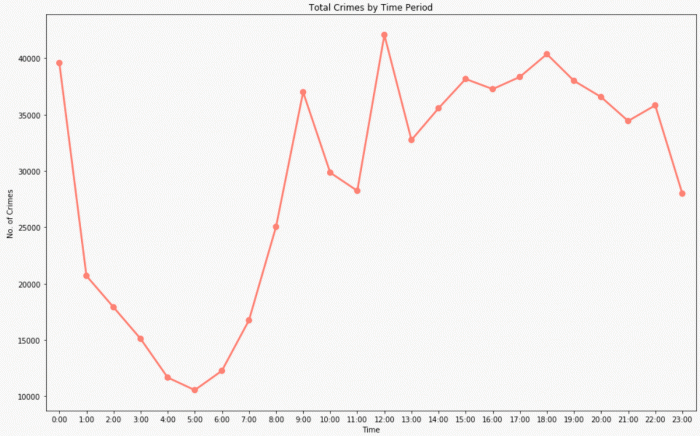
Frequency of Crimes Per Month (2012 to 2016)

From the trend above, we can see that the frequency of crimes in Chicago is decreasing, with seasonal peaks between June to August, and drops between January to March.

#### **4.3MONTHLY AND HOURLY CRIME TREND ANALYSIS**

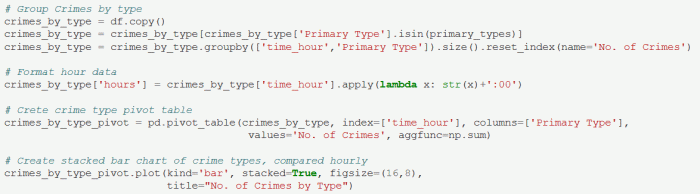
From the data above, we are going to specifically focus on the high-frequency crimes with a rising trend. (Ignoring Stalking, Obscenity, Human Trafficking, and Non-criminal offenses) We first filter the dataset of crimes, group them by the hour of the day, and subsequently plotting them based on the frequency of occurrence.

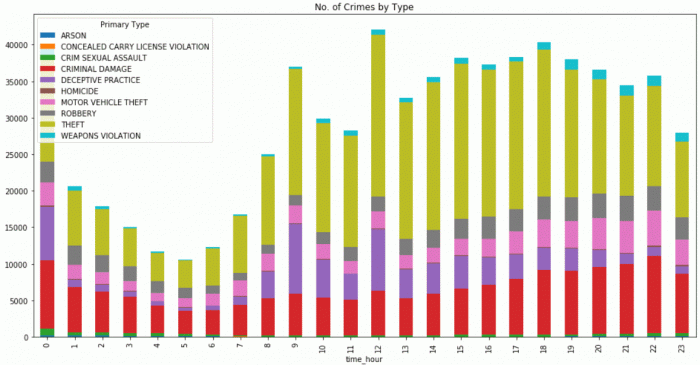




Hourly Crime Frequency

**4.4 CRIME BASED ON CRIME TYPE**



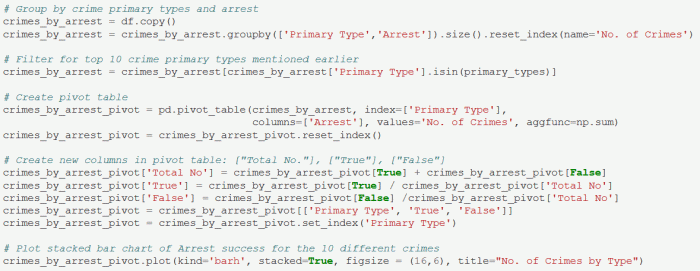


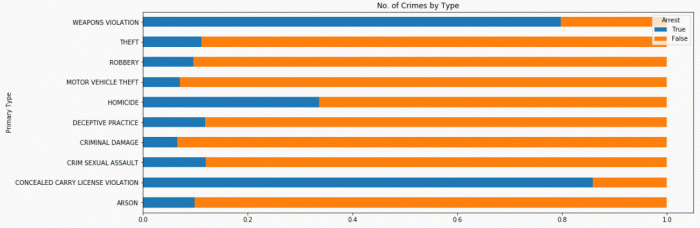
Hourly Crime Frequency by Crime Types

 the trends across the time periods, each crime type seems to have similar proportions within each hour. Theft and Criminal Damage form a significant portion of crimes committed. Deceptive practices happen more often during day-time, perhaps being linked to white-collar crimes with arrests that happen during office hours.

#### 4.5**CRIME ARREST BASED ON CRIME TYPES**

Using the same data of high-frequency crimes with rising trends, we are looking into the proportion of arrest for each crime. This time, we group the data by crime type and arrest, then we retrieve the ratio between true and false arrest.

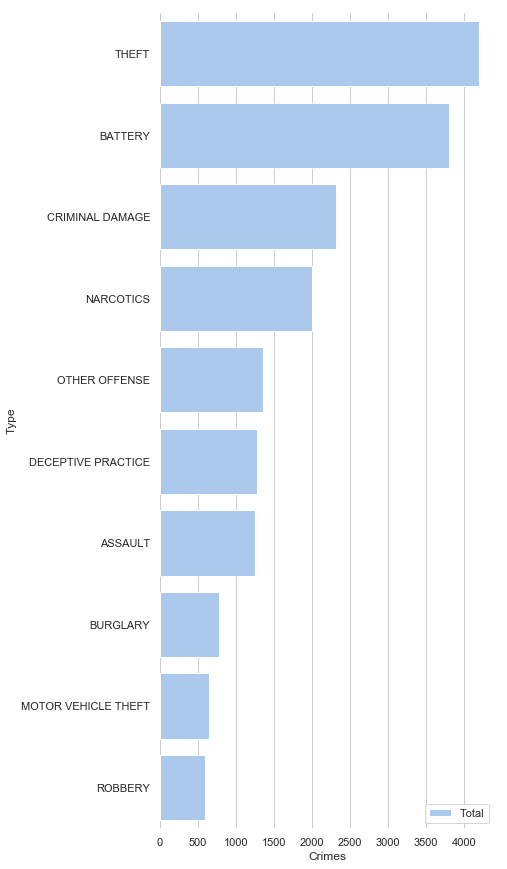




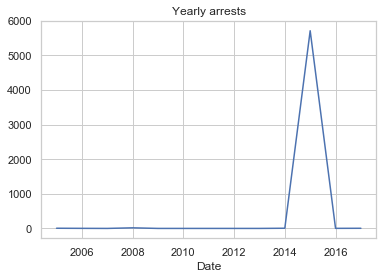
Crimes Arrest Ratio

his analysis clearly tells us that crimes involving “Weapons Violation” & “Concealed Carry License Violation” have a high arrest count. However, for the other crime types, the Chicago police department might want to look into better ways of tackling investigations. There is also a possibility of false reports and allegations.

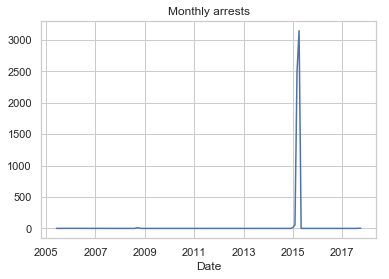
**4.6 TYPES OF CRIME OCCURRED IN CHICAGO**



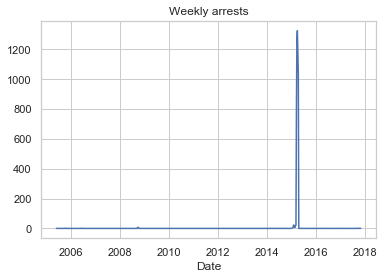
**4.7 YEARLY ARRESTS**

****

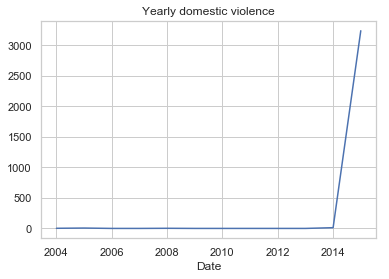
**4.8 MONTHLY ARRESTS**

****

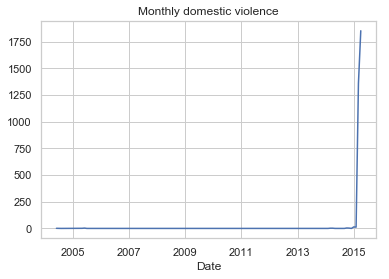
**4.9 WEEKLY ARRESTS**

****

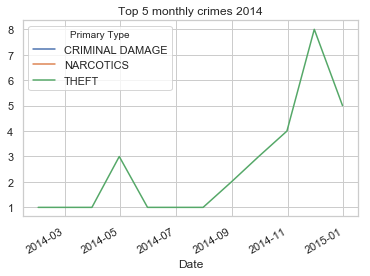
**4.10 YEARLY DOMESTIC VOILENCE**

****

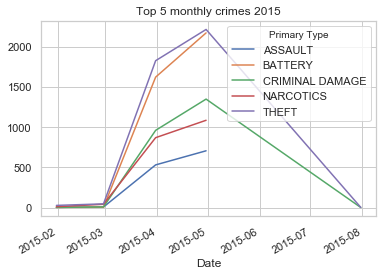
**4.11MONTHLY DOMESTIC VIOLENCE**

****

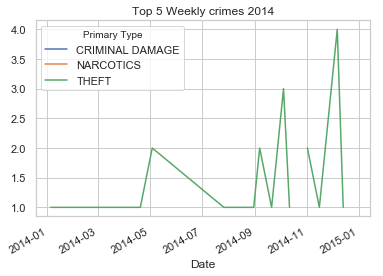
4.12 TOP 5 MONTHLY CRIMES OF 2014



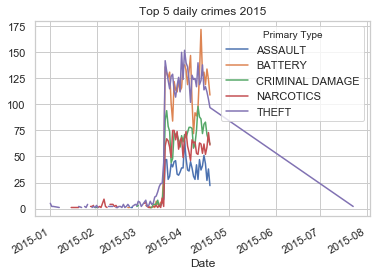
4.13 TOP M ONTHLY CRIMES IN 2015



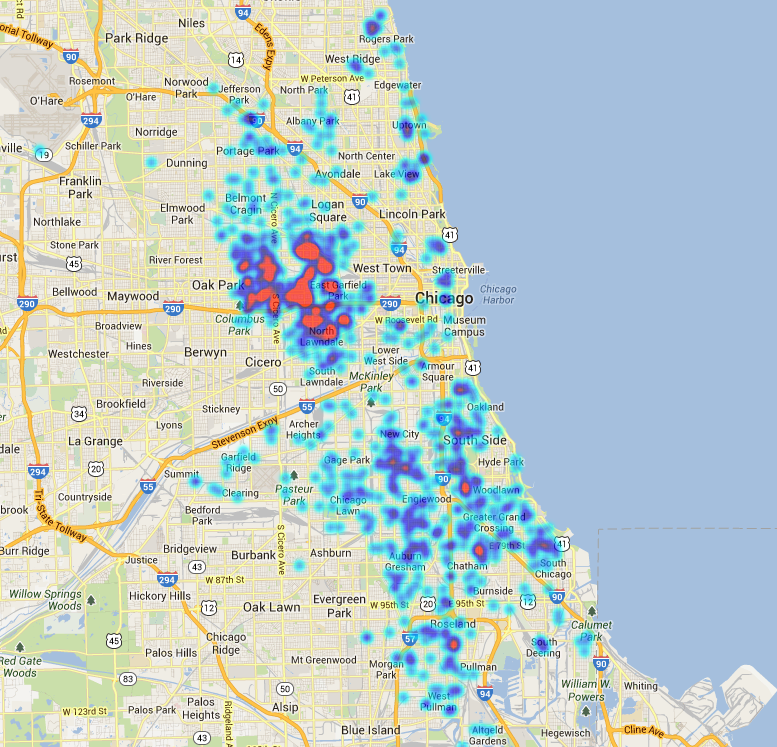
4.14 TOP WEEKLY CRIMES IN 2014



4.15 TOP WEEKLY CRIMES IN 2015



4.16 HEAT MAP SHOWING THE DANGEROUS AREAS



**5.TRACKING THE CRIMINALS**

 5.1 **Theory of OpenCV face recognizer**

computer coding facial recognition, which are similar to the steps that our brains use for recognizing faces. These steps are:

Data Gathering: Gather face data (face images in this case) of the persons you want to identify.

Train the Recognizer: Feed that face data and respective names of each face to the recognizer so that it can learn.

Recognition: Feed new faces of that people and see if the face recognizer you just trained recognizes them.

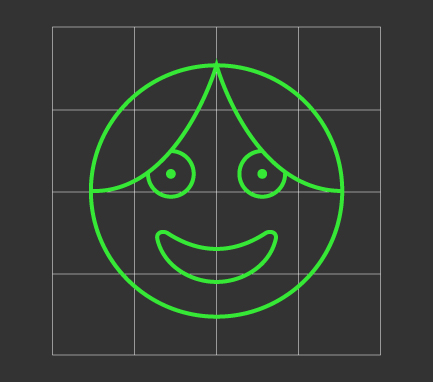
OpenCV has three built-in face recognizers and thanks to its clean coding, you can use any of them just by changing a single line of code. Here are the names of those face recognizers and their OpenCV calls:

* EigenFaces – cv2.face.createEigenFaceRecognizer()
* FisherFaces – cv2.face.createFisherFaceRecognizer()
* Local Binary Patterns Histograms (LBPH) – cv2.face.createLBPHFaceRecognizer()

**5.2 LOCAL BINARY CASCADER**

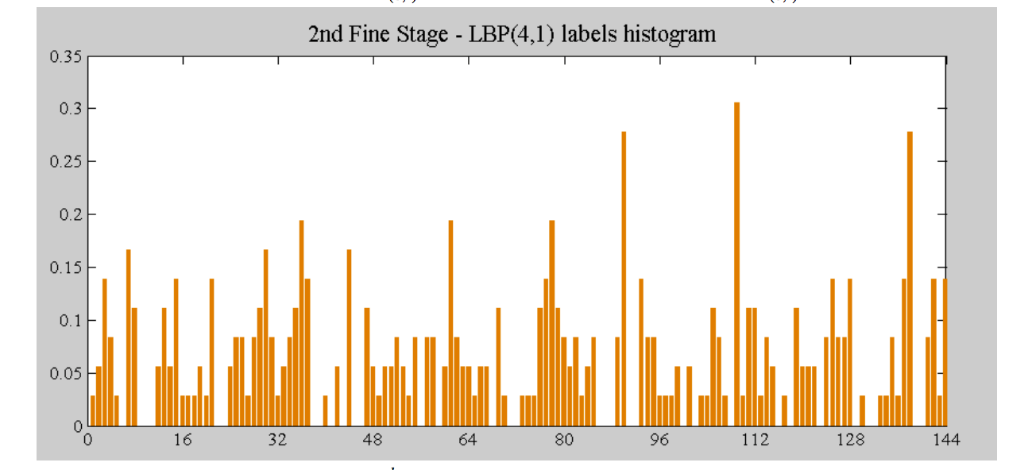
**LBP Cascade Classifier**

As any other classifier, the Local Binary Patterns, or LBP in short, also needs to be trained on hundreds of images. LBP is a visual/texture descriptor, and thankfully, our faces are also composed of micro visual patterns.So, LBP features are extracted to form a feature vector that classifies a face from a non-face.Each training image is divided into some blocks as shown in the picture below.



LBP Windows (disregard the first grader drawing)

For each block, **LBP looks at 9 pixels** (3×3 window) at a time, and with a particular interest in the pixel located in the center of the window.Then, it compares the central pixel value with every neighbor's pixel value under the 3×3 window. For each neighbor pixel that is greater than or equal to the center pixel, it sets its value to 1, and for the others, it sets them to 0.After that, it reads the updated pixel values (which can be either 0 or 1) in a clockwise order and forms a binary number. Next, it converts the binary number into a decimal number, and that decimal number is the new value of the center pixel. We do this for every pixel in a block.Then, it converts each block values into a [histogram](https://en.wikipedia.org/wiki/Histogram), so now we have gotten one histogram for each block in an image, like this:



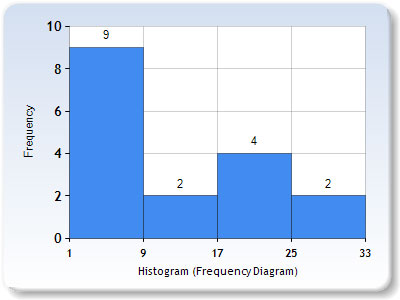
LBP Histogram. Source: López& Ruiz; Local Binary Patterns applied to Face Detection and Recognition.Finally, it concatenates these block histograms to form a one feature vector for one image, which contains all the features we are interested.

5.3**The LBPH Face Recognizer Process**

Take a 3×3 window and move it across one image. At each move (each local part of the picture), compare the pixel at the center, with its surrounding pixels. Denote the neighbors with intensity value less than or equal to the center pixel by 1 and the rest by 0.After you read these 0/1 values under the 3×3 window in a clockwise order, you will have a binary pattern like 11100011 that is local to a particular area of the picture. When you finish doing this on the whole image, you will have a list of **local binary patterns**.



LBP conversion to binary. Source: López& Ruiz; Local Binary Patterns applied to Face Detection and Recognition.Now, after you get a list of local binary patterns, you convert each one into a decimal number using [binary to decimal conversion](https://www.mathsisfun.com/binary-number-system.html) (as shown in above image) and then you make a [histogram](https://www.mathsisfun.com/data/histograms.html)of all of those decimal values. A sample histogram looks like this:

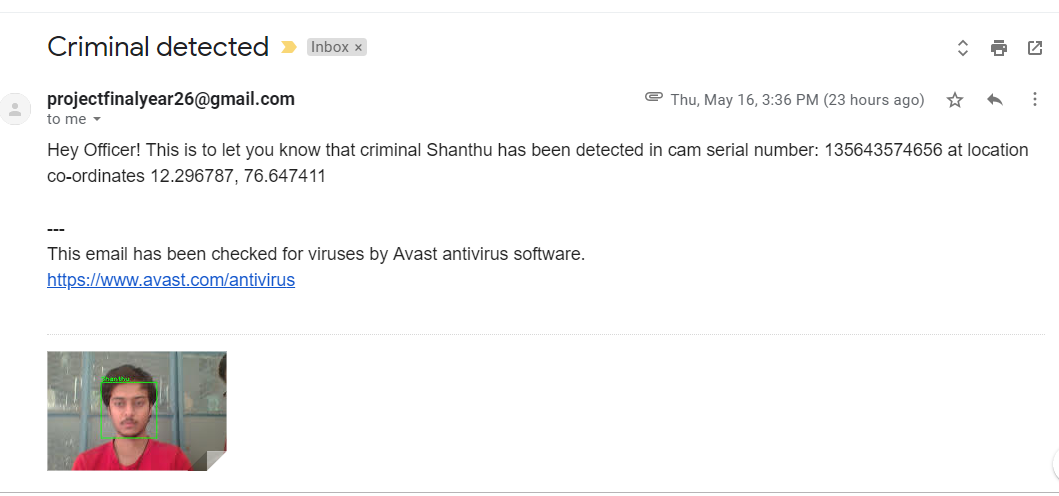


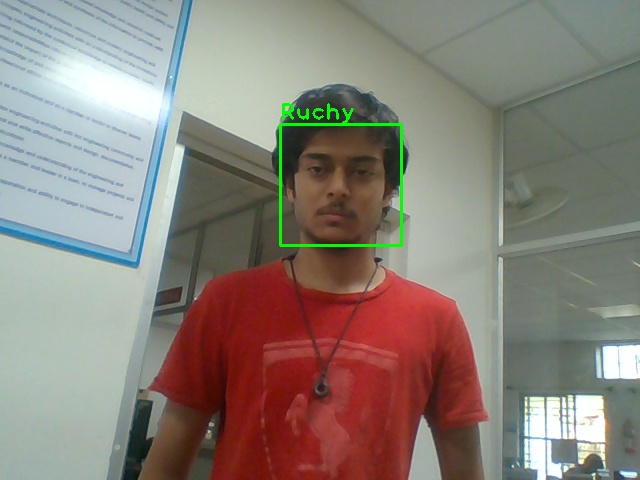
Histogram Sample.

In the end, you will have one histogram for each face in the training data set. That means that if there were 100 images in the training data set then LBPH will extract 100 histograms after training and store them for later recognition. Remember, the algorithm also keeps track of which histogram belongs to which person.

Later during recognition, the process is as follows:

1. Feed a new image to the recognizer for face recognition.
2. The recognizer generates a histogram for that new picture.
3. It then compares that histogram with the histograms it already has.
4. Finally, it finds the best match and returns the person label associated with that best match.





**CONCLUSIONS**

We believe this data analytic project give us a scientific view about the security status and crime rate of the Chicago city. According to the analysis result and visualization, we can view the most frequently occurring crimes and the frequent occurring locations where crimes happened. From these reports, the most occurred crimes were theft, battery, criminal damage and narcotics which is 65.7% of all the crimes reported. The most common locations to occur the crimes are at street, sidewalks, residence, an apartment which are where people are mostly at. We specifically looked into certain crime types to view how they have changed over the years, such as theft, homicide, and sexual crimes. Even though there were a lot of reported crimes in Chicago each year, the arrest rate was not even as high as 50% for each year letting us believe that Chicago’s police arrest or investigation methods were not effective enough. We believe if our data analytics can give us all these information about the security status of the Chicago city, a bigger data analytics project will provide much more valuable information which can be used as a powerful source for taking wise actions that increases the security status of our cities

**FUTURE SCOPES**

**REFEREENCES**

* Ahonen, Timo, Abdenour Hadid, and Matti Pietikainen. “**Face description with local binary patterns: Application to face recognition**.” IEEE transactions on pattern analysis and machine intelligence 28.12 (2006): 2037–2041.
* Ojala, Timo, Matti Pietikainen, and Topi Maenpaa. “**Multiresolution gray-scale and rotation invariant texture classification with local binary patterns**.” IEEE Transactions on pattern analysis and machine intelligence24.7 (2002): 971–987.
* Ahonen, Timo, Abdenour Hadid, and Matti Pietikäinen. “**Face recognition with local binary patterns**.” Computer vision-eccv 2004 (2004): 469–481.
* LBPH OpenCV: <https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms>
* Local Binary Patterns: <http://www.scholarpedia.org/article/Local_Binary_Patterns>
* [1]Currie32, 2017. Crimes in Chicago. RSNA Pneumonia Detection Challenge | Kaggle. Available at: <https://www.kaggle.com/currie32/crimes-in-chicago>[Accessed December 14, 2018].
* [2]Anon, Wrangler is an interactive tool for data cleaning and transformation. Data Wrangler. Available at: <http://vis.stanford.edu/wrangler/> [Accessed December 7, 2018].
* [3]Anon, Tutorials. pandas: powerful Python data analysis toolkit — pandas 0.23.4 documentation. Available at: <http://pandas.pydata.org/pandas-docs/version/0.15/tutorials.html> [Accessed December 14, 2018].
* [4]Anon, Welcome! openrefine.github.com. Available at: <http://openrefine.org/> [Accessed December 14, 2018].
* [5]Stevestein, SQL Server Management Studio (SSMS). Microsoft Docs. Available at: <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-2017> [Accessed December 14, 2018].
* [6]Anon, MapReduce Tutorial. Apache Hadoop. Available at: <https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html> [Accessed December 14, 2018].
* [7] Anon, Spark Overview. Apache Spark™ — Unified Analytics Engine for Big Data. Available at: <https://spark.apache.org/docs/latest/> [Accessed December 14, 2018].
* [8]Anon, Google. Available at: <https://developers.google.com/chart/interactive/docs/spreadsheets>[Accessed December 14, 2018].
* [9]Anon, 2018. crimeanalyze.firebaseapp.com. Available at: <https://crimeanalyze.firebaseapp.com/> [Accessed December 14, 2018]