International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-6, April 2020

Web Superintendence

Upendra Kumar Tiwari, Abhay Chauhan, Devraj Kumar, Ayush Gupta

Abstract: ***Over the past two decades, the internet has exponentially expanded to every part of the world. It has made every single individual as its either consumer or even a producer. And with the evolution of different social networking sites, it has even made individuals upload their personal content over the internet. This personal information compromises information ranging from even the smallest detail ranging from his birthdate to even the place he is traveling to. Uploading personal information over the internet has leveraged various opportunities, it could either be business intelligence or even for the purpose of overseeing a person with antisocial behavior.***

Keywords: Open source Data, surveillance, data mining, Web Scraping, Web Crawler, Data Forensic

I. INTRODUCTION

Present ways to oversee somebody generally require them to be physically present to observe or telephone tapping. These techniques prove to be either costly or dangerous. Also for superintending someone by ways such as phone tapping require permissions from higher authorities and is highly questionable in many ways.

Although other methods are also available but have been ignored even being more feasible and foolproof. Every person has data about his activities in the public domain and is easily accessible via the internet. It may be his public profiles or maybe news websites that can tell a lot about a person. Facebook has approximately 2.5 billion active users and Twitter has around 126 million active users. On Facebook itself, around 55 million status updates are made every day. This provides a very large database for superintending someone, no matter what use case it is for. As these status updates involve the feelings and sentiments of the subject along with the places he visits and activities he is or has been involved in.

Even if the internet is surfed for collecting data related to an individual, A typical scenario is to look manually through all the links provided by the search engines for specific key terms. These links are not specifically related to the desired search and very little information of interest is found.

Revised Manuscript Received on April 30, 2020.
***** Correspondence Author
Upendra Kumar Tiwari* Assistant Professor, Department of Computer Science and Engineering, ABES Institute of Technology, Ghaziabad, India.
Abhay Chauhan, Department of Computer Science and Engineering, ABES Institute of Technology, Ghaziabad, India.
Devraj Kumar, Department of Computer Science and Engineering, ABES Institute of Technology, Ghaziabad, India.
Ayush Gupta, Department of Computer Science and Engineering, ABES Institute of Technology, Ghaziabad, India.

In contrast, the present technique generally aims at automating the process of surveillance and keeping track of the subject's real-world activities.

The approach is to parse the internet in a depth-first search manner that involves providing basic details about the subject such as name, photograph and any other specific terms such as workplace or residential region. All the search results are being parsed, importantly social media webpages are focussed to extract images.

Further, these images are compared to the image initially provided to the system using the HoG (Histogram of oriented gradients) algorithm. The advantage of using a face recognition algorithm is that it can provide high accuracy in comparing faces when both images look really different from an ordinary person's vision. Once the faces in both images are confirmed to be compared as of the same person's, the profile is assumed of the subject. Further profiles are monitored and all updates are being stored.

In a real-world system like this could be used for keeping track of one's activities without making any efforts. This practice of monitoring suspects can result in a drastic drop in the number of offensive incidents.

II. RELATED WORK

The web is full of data now, this data specifically over social media sites can impact a person in both the way a positive or a negative one. In recent times we have seen numerous cases where hate speech or false information over the internet led to mass riots and affected a numerous count of people. To tackle similar situation the following measure is usually initiated under the influence of governing bodies

III. THE MANUAL APPROACH

The manual approach is the simplest method, which the governing bodies tackle such information, though this method is the easiest but most time consuming, the person responsible has to make manual searches over various social media platforms, if there's any such case or not.

IV. THE CROWDSOURCED INFORMATION

With the increase in user and advanced tech, many people aim at making this world a better place. Under this method a group of people voluntarily usually report if they found anything suspicious over a common platform, which is under the control of the governing body.

And once the information is over the platform further suitable action is taken in the similar direction. This method has a comparatively high rate of execution then the manual approach, but the false reported information index is comparatively high in the same.

V. OUR APPROACH

Our approach is mainly focused on the application of web crawler and web scraping.

At the initial stage the process needs some manual input regarding the key words he wants the result for. Once the keywords are fed into the program. A headless browser is initiated, and the parameters are passed on in the search bar of the social media platforms we want to search upon. Upon result, the page will be parsed as an HTML document, and then the crawler will look for the desired css id and xpath and evaluate each one of them. And will render each as desired. This process is fast enough as multiple social media platforms can be traversed at the same instance. And to add accuracy in result the scrapped results, which are in the format of images/videos are treated as another input set which needs to be tested against certain test cases, the test cases are validated using the image processing approach which enables us to provide us with a better accuracy rate than ever before.

```
initate headless browser
  result = parse.(inputparameter.html)
   {
       result_1 = result.css("selector")
        for (each selector)
        {
            data = get(link.content)
        }
       result_2 = result.xpath("xpath")
        for (each xpath)
        {
            data = get(link.content)
        }
       result_3 = result.combo("value")
        for (each value)
        {
            data = get(link.content)
        }
   }

#input
inputparameter = the key words regarding
                 which you want data
selector = css id of element
xpath = xpath of element
value = combination of xpath and
        css of element
```

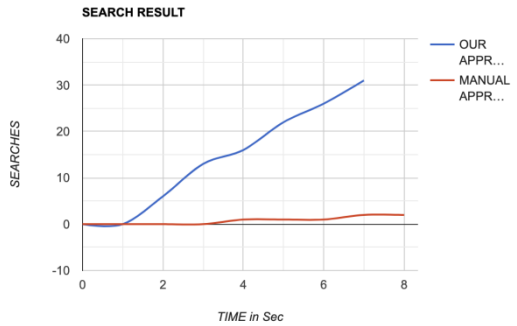
VI. ALGORITHM FOR CRAWLER

ALGORITHM FOR IMAGE PROCESSING

```
Histogram of Oriented Gradients

* HoG descriptor are used for edge detection by gradient calculation and histograms of gradient with
magnitudes as weights.
* [-1 0 -1] Kernel is used for gradient magnitude and orientation calculation.
* Gradients' range == [0, 180]
* 32 * 32 size of images are used.
* Images are read in greyscale.

ALGORITHM

1. convert_image_into_array
       return image_array
2. create_gradient_array
    image_array = power(image_array, 2.5) // Gamma Correction
    image_array = (image_array - mean(image_array)) / standard_deviation(image_array)

    maximum_height = 32
    maximum_width = 32

    gradient = zero_matrix(maximum_height, maximum_width)
    magnitude = zero_matrix(maximum_height, maximum_width)

    for h,row in enumerate(image_array):
        for w, val in enumerate(row):
            if h-1!=0 and w-1!=0 and h+1<maximum_height and w+1<maximum_width:
                dy = image_array[h+1][w]-image_array[h-1][w]
                dx = row[w+1]-row[w-1]+0.0001
                gradient[h][w] = arctangent(dy/dx)*(180/pi)
                if gradient[h][w]<0:
                    gradient[h][w] += 180
                magnitude[h][w] = square_root(dy*dy+dx*dx)

    return gradient,magnitude
3.calculate_histogram(arr,weight)
    bns_range = (0,180)
    bins = 8
    hist = Histogram(arr,bins,bins_range,weight)

    return hist
4. create_HoG_features(gradient_array, magnitute_array)
    incr = [8,8]
    cell = [8,8]

    maximum_height = ((gradient_array.shape[0] - cell[0]) / incr[0] + 1)
    maximum_width = ((gradient_array.shape[1] - cell[1]) / incr[1] + 1)

    cell_array = []
    w = 0
    h = 0
    i = 0
    j = 0

    while i<maximum_height: //Creating 8X8 cells
        w = 0
        j = 0

        while j<maximum_width:
            for_hist = gradient_array[h:h+cell[0],w:w+cell[1]]
            for_wght = magnitute_array[h:h+cell[0],w:w+cell[1]]

            val = calculate_histogram(for_hist,for_wght)
            cell_array.append(val)
            j += 1
            w += incr[1]

        i += 1
        h += incr[0]

    cell_array = reshape(cell_array,(max_h, max_w, bin_num))

    block = [2,2] //normalising blocks of cells

    maximum_height = int((max_h-block[0])+1)
    maximum_width = int((max_w-block[1])+1)
    block_list = []
    w = 0
    h = 0
    i = 0
    j = 0

    while i<maximum_height:
        w = 0
        j = 0

        while j<maximum_width:
            for_norm = cell_array[h:h+block[0],w:w+block[1]]
            magnitude = np.linalg.norm(for_norm)
            arr_list = (for_norm/magnitude).flatten().tolist()
            block_list += arr_list
            j += 1
            w += 1

        i += 1
        h += 1

    return block_list //returns a vector array list of 288 elements
5. apply_HoG(image_array)
    grad,mag = create_gradient_array(image_array)
    hog_feat = create_HoG_features(grad,mag)
    hog_feat = convert_into_array(hog_feat)
    hog_feat = expand_dimensions(hog_feat, axis = 0)

    return hog_feat
```

VII. COMPARATIVE STUDY

Above illustrated line graph displays how our proposed algorithm stands out when compared to the traditional manual approaches. At the initial time frame both approaches have 0 results as the time is consumed in loading the page.



Once the required page is loaded the algorithm starts fetching results at a significant rate while in the manual approach the operator is still busy with the task to find where the suitable information is located. Also one of the considerable facts is that manual approaches are error prone while the algorithmistic approach are not.

VIII. RESULT

The result of data analysis shows a clear idea, that the traditional methods are very slow in producing the result while our approach is significantly fast. As you can visualize with graph that in the intal time frame both perform the same because within this time frame all the data is gathered and with the increase in time while the traditional approach analyzes just a certain number of data, the automatic approach is almost 10x ahead, with much better accuracy.

IX. CONCLUSION

The purpose of our work was to introduce a much faster and efficient way in order to monitor the data over the web. To compete with the web, using a manual approach is not at all advisable. So to overcome the situation we have proposed an algorithm which is fast, efficient and produces satisfactory results in the meantime.

The application of or work is not limited just to monitoring certain keywords but it can also work in real time monitoring of suspicious people with suitable modification.

REFERENCES

1. Laxmi Prashanthi Karnati BVNSS Pavan, Prof. Manikandan, Vellore Institute of Technology, November 2016, IJIRT (Volume 3 Issue 6)
2. "Can Scraping Non-Infringing Content Become Copyright Infringement... Because Of How Scrapers Work? | Techdirt". *Techdirt*. 2009-06-10. Retrieved 2016-05-24.
3. ^ "Facebook v. Power Ventures". *Electronic Frontier Foundation*. Retrieved 2016-05-24.
4. ^ "UDSKRIFT AF SØ- & HANDELSRETTENS DOMBOG" (PDF) (in Danish). bvhd.dk. 2006-02-24. Archived from the original (PDF) on 2007-10-12. Retrieved 2007-05-30.
5. ^ "High Court of Ireland Decisions >> Ryanair Ltd -v- Billigfluege.de GMBH 2010 IEHC 47 (26 February 2010)". British and Irish Legal Information Institute. 2010-02-26. Retrieved 2012-04-19.
6. ^ Matthews, Áine (June 2010). "Intellectual Property: Website Terms of Use". *Issue 26: June 2010*. LK Shields Solicitors Update. p. 03. Retrieved 2012-04-19.
7. ^ National Office for the Information Economy (February 2004). "Spam Act 2003: An overview for business". Australian Communications Authority. p. 6. Retrieved 2017-12-07.
8. ^ National Office for the Information Economy (February 2004). "Spam Act 2003: A practical guide for business" (PDF). Australian Communications Authority. p. 20. Retrieved 2017-12-07.
9. ^ Mayank Dhiman Breaking Fraud & Bot Detection Solutions *OWASP AppSec Cali' 2018* Retrieved February 10, 2018.

AUTHORS PROFILE

Upendra Kumar Tiwari is working as an Asstt. Professor in ABES Institute of Technology, Ghaziabad. He has more than 13 year of experience in academic. he has published 3 books. His areas of interest are Data Science, Machine Learning.

Abhay Chauhan is a final year student pursuing B.tech from Computer Science and engineering branch at ABES Institute of Technology, Ghaziabad. He has served the role of Explore ML Facilitator.

Devraj Kumar is a final year student pursuing B.tech from Computer Science and engineering branch at ABES Institute of Technology, Ghaziabad. He has a winning streak in Smart India Hackathon for the last 2 years.

Ayush Gupta is a final year student pursuing B.tech from Computer Science and engineering branch at ABES Institute of Technology, Ghaziabad.He loves designing algorithms to solve real world problems efficiently.