IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Facial Recognition with Web Crawler Application[1]

Ashiq Badusha B and Dr.Akashdeep Bharadwaj

University of Petroleum and Energy Studies, Dehradun http://www.Upes.com

1 Abstract

The widespread use of social media, with over half the world's population now active on these platforms, has made it easier than ever to search for individuals by using their profiles. A new application that integrates facial recognition technology with web scraping could be a valuable tool for this digital age. There are already established methods for identifying similar faces using features and deep learning techniques, and a web crawler can gather information from social media websites to create a dataset for comparison and analysis.
Face detection and web scraping are two of the most widely studied subjects in the realm of cybersecurity, and an application that combines these two techniques could help build a resource of information and images from public platforms. This resource could then be used with a face recognition system to search for individuals on social media by image. The application could also be useful for law enforcement agencies and investigation units, helping them to identify people from a caricature sketch or facial measurements and gathering information from various resources, including social media.

Keywords: Facial Recognition · Web Crawler · Face search engin.

2 Introduction

As technology continues to advance and the number of internet users grows, so does the need for security. According to a 2022 study, 467 million people in India and 4.66 billion worldwide are using social media, making it a critical area for security. As a result, face recognition has become a popular research area in response to the growing demand for security and the increasing use of mobile devices. Facial recognition is being used for various purposes, including unlocking smartphones, social media networks, security systems, authentication, and surveillance. It is becoming a more viable alternative to passwords and fingerprint scanners in areas such as offices, laptops, phones, ATMs, and other devices that require access control. Facial recognition can also be used to identify similar images and gather information about a face from available sources. For example, Facebook allows users to tag friends in photographs using facial recognition.

Web scraping, or web harvesting, is a technique used to extract data from the internet and store it for further analysis or use. It is similar to web crawling and often involves the use of software that collects data from websites and stores it in a database for later processing. Web scraping can be performed manually or using automated bots and involves acquiring data from the web and then extracting specific information from the acquired data.

3 Related Work

As the web grows larger, accessing web pages becomes increasingly challenging without having the direct address. Search engines address this problem by using a process called web crawling (Castillo, 2004). This algorithm consists of three steps: retrieving a webpage, extracting all linked URLs, and repeating the process for previously unseen URLs. The goal of web crawling is to scan and index a collection of websites so that they can be searched effectively. The use of web scrapers has proven to be an effective tool for law enforcement, particularly in the investigation of human trafficking, as it allows them to identify potential trafficking activities in the early stages of recruitment (McAlister, 2015). Nawaf Hazim Barnouti (2016) presented an automatic face recognition system that is based on exterior-

[1] Supported by UPES.

based techniques, using the Viola-Jones method for face detection and gathering information from various databases. The system determines the similarity between two images by calculating the square Euclidean distance between them. Ningthoujam Sunita Devi proposed a face recognition technique that combines information theory and machine learning. The method involves two stages, the first being the extraction of features using principal component analysis, and the second being recognition using a feed forward backpropagation neural network. The methodology was tested using the Oracle Research Laboratory face database, which contains 400 images, and showed promising results in overcoming the challenges of face recognition such as variations in recorded images due to changes in pose, lighting, expressions, hairstyles, glasses, and other factors.

4 Methodology

4.1 Introduction

Facial recognition has recently become a popular research area due to the increasing demand for security and the advancements in mobile technology. The purpose of facial recognition is to identify and retrieve information about a face from available sources. An example of this can be seen in the use of facial recognition on social networking platforms such as Facebook where users are encouraged to tag friends in photos. There are two main approaches to facial recognition, feature-based and view-based, which use spatial features and photometric methods respectively.

Web scraping, also known as web harvesting, is a technique used to extract data from the World Wide Web and save it for future analysis. A web scraper is software that collects data from webpages and stores it in a database for later processing. This process is done using the Hypertext Transfer Protocol or through a web browser, either manually or with automated bots. Web scraping consists of two main parts: acquiring resources from the web and extracting specific desired information from the acquired data.

4.2 Need of Project

The combination of face detection and web scraping is a popular area of research in the field of cyber technology. The goal of this research is to develop an application that can gather information and images from public platforms and create a dataset that can be used with a face recognition system to search for similar images on social media. This technology has the potential to be useful for cyber cells, police departments, and other investigative agencies for identifying individuals based on their facial features and gathering information from social media or other sources. The facial features are extracted from an input image and used to search for similar images in the dataset or resource, which can provide additional information. Web scraping is also used to gather data and create the dataset or resource. The image search feature can improve the performance of the application, as well as find profile information on social media platforms like Facebook and Instagram.

4.3 Scope of the Project

The learning of facial recognition and web scraping can be divided into objectivesbased projects. The facial recognition project aims to improve understanding in new technologies, models, design, algorithms, and more, by comparing and finding similarities between images. On the other hand, the web scraping project involves gathering information from various sources, such as websites and social media, and creating a database or resource that can be used for future development and manipulation. The project also involves preparing and simplifying the data, and using it with different models and techniques.

4.4 Methodology

To summarize, web scraping is a process of extracting information from websites by downloading webpages and parsing the HTML code to extract the desired information. A web scraper is an automated tool that carries out these steps, which includes using the requests library to download webpages, using the Beautiful Soup library to parse the HTML source code, building the scraper components, compiling the extracted information into Python lists and dictionaries, converting the dictionaries into Pandas dataframes, and writing the information into a final CSV file. A web scraper is different from a web crawler, as the latter only crawls and indexes webpages, while a web scraper extracts data. The process of web scraping involves obtaining resources from websites and then extracting the desired information from those resources. This is done by creating an HTTP request to access resources from the target website. The website responds to the request by delivering the resource back to the web scraping software. The resource can be in different formats like HTML pages, XML, JSON data feeds, or multimedia files. The web scraping software must have two components: one for making HTTP requests and another for parsing and extracting information from the raw HTML code. The extracted information is then formatted and organized in a systematic manner to make it usable. The libraries such as Urllib2 and selenium are used to make HTTP requests, while BeautifulSoup and Pyquery are used to parse the HTML code and extract information. t also provides a clean and intuitive syntax for

navigating and manipulating the data, allowing developers to select and extract data from XML and HTML documents in an efficient manner.

In conclusion, web scraping requires two main components, a method of sending HTTP requests and a tool for extracting information from the returned data. The choice of tools and libraries will depend on the specific requirements of the project, but some popular choices include Urllib2, Selenium, BeautifulSoup, and Pyquery. The end goal is to obtain and process web data, whether it be in the form of HTML, XML, or other multimedia formats, in order to extract the desired information and use it for various purposes such as data analysis, data storage, or information retrieval. It is important to note that there are numerous web scraping solutions available, each with its own strengths and weaknesses. Nutch, Scrapy, and Import.io are some of the popular web scraping tools that are used for extracting data from websites. Nutch is a reliable and scalable web crawler that supports robots.txt rule, parallel harvesting, fine-grained configuration, and machine learning. Scrapy is a Python-based reusable web crawling framework that expedites large crawling project construction and scaling. Import.io, on the other hand, is a web-based crawler that offers a graphical interface, making it easier for non-programmers to extract web contents. The extracted data can be exported in various formats, such as CSV, JSON, and XML, and can be stored on a dedicated cloud server. However, while a web-based crawler with a graphical interface makes it easier to harvest and visualize real-time data streams, it may have difficulties handling massive datasets.

It's worth noting that the accuracy of face recognition systems depends on several factors, including the quality of the images used for training and the size of the training dataset. Additionally, face recognition algorithms may be affected by factors such as pose, lighting, and facial expressions, which can impact their performance. To improve accuracy, it's important to use a diverse and representative dataset for training and to continuously evaluate and improve the system through ongoing research and development. Face recognition system encompasses three main phases which are face detection, feature extraction, face recognition.

1. *Face Detection: Face acquisition and localisation from an image is detecting with CNNAlgorithm. Pre-processing of human faces are separated from the objects present in an image.*

2. *Feature Extraction: From the detected face we are extracting the featuresthrough multi-tasking cascaded convolutional neural network. In mtCNN, In the first stage, it produces candidate windows quickly through a shallow CNN. Then, it refines the windows to reject a large number of non-faces windows through a more complex CNN. Finally, it uses a more powerful CNN to refine the result and output facial landmarks positions.*

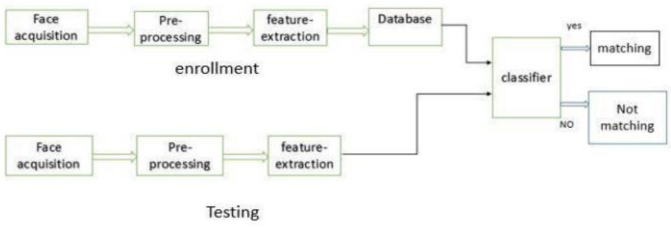

Fig.1. Facial recognition analyser

3. *Face Recognition: The extracted features are fed to the classifier which recognizes or classifies by using Machine Learning algorithm. The classifier compares the test image with the images saved in the database can be done with supervised machine learning classifier. scikit-learn python machine learning library and Euclidean distance used for this operation*

4.5 Data Analyse

In the process of facial recognition using multi-task cascaded convolutional neural network (MTCNN), the original image is transformed into a different range, referred to as an image pyramid. The first model in the network identifies the face region of the image, followed by filtering and refining the image through subsequent models. The final stage of the process involves proposing facial landmarks on the image. MTCNN has pre-trained models that are available as opensource, which can be used for direct applications. The architecture of MTCNN is implemented using openCV and Tensorflow. The feature vectors in a facial recognition system are a set of numerical values, often referred to as embeddings or bottleneck features. They are derived from the image through the use of convolution and pooling layers in a CNN, which extract the most important and relevant features of the image. The CNN is trained to adjust these values so that faces belonging to the same person are closer together in terms of Euclidean distance, while faces from different people are farther apart. This characteristic is essential for successful classification, especially in unsupervised scenarios when labeled data is limited. The scikit-learn Python machine learning library can be used to perform similarity searches, which involves finding the nearest neighbors of the query features stored during the image input. Both supervised and unsupervised nearest neighbors-based learning techniques can be implemented using the sklearn.neighbors platform. Other learning techniques such as spectral clustering and manifold learning are

also based on the foundation of unsupervised nearest neighbors. Classification for data with discrete labels and regression for data with continuous labels are the two forms of supervised neighbours-based learning. The idea behind nearest neighbour approaches is to select a set number of training samples that are geographically closest to the new point and then estimate the label based on them. A user-defined constant (k-nearest neighbour learning) or a variable dependent on the local density of points are both possible for the number of samples (radius-based neighbour learning). In general, the distance can be measured in any metric unit; the most popular option is the conventional Euclidean distance. Ball tree query time increases as about O[Dlog(N)]

$$dist(q, img) = ||q - img||_2 = \sqrt{\sum_{i=1}^{n}(q_i - img_i)^2}$$

Fig.2. Euclidean distance.

while KD tree query time changes with D in an unpredictable manner. The cost is roughly O[Dlog(N)], and the KD tree query can be highly effective for small D (less than 20 or so). The cost rises to almost O[DN] for larger D, and the overhead of the tree structure can make queries slower than using brute force. Data structure has little impact on the speed of brute force queries. Data structure can have a significant impact on the query timings for Ball trees and KD trees. Faster query times are typically the result of sparser data with a lower intrinsic dimensionality. There are two distinct neighbours regressors that scikit-learn implements: KNeighborsRegressor, where k is an integer value supplied by the user, executes learning based on each query point's k nearest neighbours. The RadiusNeighborsRegressor carries out learning based on the neighbours that are within a set radius r of the query point, where r is a floating- point value that the user specifies. For the web crawler, In terms of data extraction, Beautiful Soup is made for XML and HTML document scraping. The framework for dissecting an HTML file and extracting needed information using lxml or html5lib is provided, and Beautiful Soup can automatically detect the encoding of the data that is being processed and convert it to a client-readable encode. Similar to Jquery, Pyquery offers a set of functions for parsing XML documents. However, Pyquery only supports lxml for quick XML processing, unlike Beautiful Soup. Some of the numerous online scraping solutions, like Nutch or Scrapy, are designed to automatically recognise the data structure of a website, while others, like Import.io, offer a web-based graphic interface that does away with the necessity for manually written webscraping code.

4.6 Proposed Work

Facial Recognition:

1. *Face Detection: Face acquisition and localisation from an image is detecting with CNNAlgorithm. Pre-processing of human faces are separated from the objects present in an image.*

2. *Feature Extraction: From the detected face we are extracting the featuresthrough multi-tasking cascaded convolutional neural network. In mtCNN, In the first stage, it produces candidate windows quickly through a shallow CNN. Then, it refines the windows to reject a large number of non-faces windows through a more complex CNN. Finally, it uses a more powerful CNN to refine the result and output facial landmarks positions.*

3. *Face Recognition: The extracted features are fed to the classifier which recognizes or classifies by using Machine Learning algorithm. The classifier compares the test image with the images saved in the database can be done with supervised machine learning classifier. scikit-learn python machine learning library and Euclidean distance used for this operation.*

Web Scraping:

1. *Download the webpages(social media sites) using requests.*

2. *Parse the HTML source code using Beautiful Soup library and extract thedesired information*

3. *Building the scraper components*

4. *Compile and extracted information into python list and dictionaries*

5. *Convert the dictionaries into Pandas Data frames*

6. Writing information into final CSV file

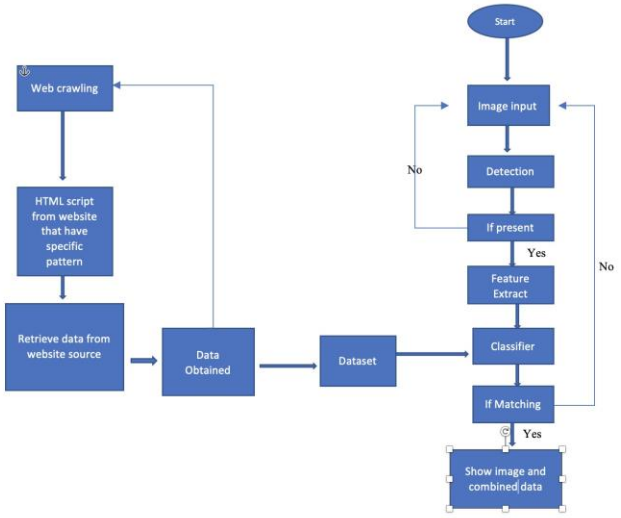

Fig.3. Flow chart of the proposed work

5 Results

The given model of MTCNN after configured and loaded, implementation of the given input image detect the face directly with the mesurements of the face with dictionary of left eye, right eye, nose, mouth left and mouth right. Also. It save the prediction confidence probability.The rebuild image is used for the analysis and detection as per MTCNN model. also the output of the model that contain the extracted feature measurement as dictionary. those measurement used to find the similar face from the web scraper dataset and get the similar face and related data

6 Conclusion

The combination of face detection and web scraping technology has the potential to provide a comprehensive resource of images and information from public platforms. This resource can be used in conjunction with a facial recognition system to search for individuals on social media using an image. The application can be highly beneficial for law enforcement agencies and investigative units, as they can identify individuals from a sketch or facial measurements and gather information from various sources, including social media.

References

1. Priyanka Dhoke, M.P. Parsai, —A MATLAB based Face Recognition using PCAwith Back Propagation Neuralnetwork‖, 2014.
2. Nawaf Hazim Barnouti ,Sinan Sameer Mahmood Al- Dabbagh ,Wael Esam Matti,Mustafa Abdul Sahib Naser, —Face Detection and Recognition Using Viola-Jones with PCA-LDA and Square Euclidean Distance‖ International Journal of Advanced Computer Science and Applications, Vol. 7, No. 5, 2016.
3. Paul Viola, Michael Jones, —Rapid object Detection using a Boosted cascade ofsimple features‖, Conference on Computer vision and Pattern recognition 2001
4. Maliha Khan; Sudeshna Chakraborty; Rani Astya; Shaveta Khepra 'Face Detectionand Recognition Using OpenCV : Publisher: IEEE'
5. Idelette Laure Kambi Beli and Chunsheng Guo, —Enhancing Face IdentificationUsing Local Binary Patterns and K-Nearest Neighbors‖, Journal of Imaging, 2017.
6. Panchakshari P, Dr. Sarika Tale, —Performance Analysis of Fusion method forEAR Biometrics‖. IEEE Conference on recent trends in Electronics, Information and Communication, May 2016.
7. Ningthoujam Sunita Devi, K. Hemachandran, —Face Recognition Using PrincipalComponent Analysis‖, International Journal of Computer Science and Information Technologies, Vol. 5 (5), 2014,pp 6491-6496.
8. Albert, R., Jeong, H. and Barabási, A. (1999) Internet: Diameter of the world-wideweb, Nature, 401(6749), pp. 130-131. doi: 10.1038/43601.
9. Broder, A.Z., Najork, M. and Wiener, J.L. (2003) Efficient URL caching for worldwide web crawling. , Budapest, Hungary. 20-24 May 2003. New York, NY, USA:
ACM, pp. 679.
10. Madhusudan, P.A. and Lambhate Poonam, D. (2017) Deep Web Crawling Efficiently using Dynamic Focused Web Crawler, International Research Journal of Engineering and Technology (IRJET), 04(06), pp. 3303.
11. McAlister, R. (2015) Webscraping As an Investigation Tool to Identify PotentialHuman Trafficking Operations in Romania. , Oxford, United Kingdom. 28-01 July 2015. New York, NY, USA: ACM, pp. 2.

12. Zhao, F., Zhou, J., Nie, C., Huang, H. and Jin, H. (2016) SmartCrawler: A Twostage Crawler for Efficiently Harvesting Deep-Web Interfaces, IEEE transactions on services computing, 9(4), pp. 608-620. doi: 10.1109/TSC.2015.2414931.
13. Scrapy (2017) Architecture overview.Available at: https://doc.scrapy.org/en/1.5/topics/architecture.html