SeekSuspect : Retrieving Suspects from Criminal Datasets using Visual Memory

Aayush Jain[1*], Meet Shah[1*], Suraj Pandey[1*], Mansi Agarwal[1*], Rajiv Ratn Shah[1], Yifang Yin[2]
[1]Indraprastha Institute of Information Technology, Delhi, India
[2] National University of Singapore, Singapore
{aayush18122, shah18046, suraj18025, mansia, rajivratn}@iiitd.ac.in, yifang@comp.nus.edu

ABSTRACT

It is crucial for the police department to automatically determine if suspects are present in the criminal database, sometimes based on the informant's visual memory alone. FaceFetch [15] is a state-of-the-art face retrieval system capable of retrieving an envisioned face from a large-scale database. Although FaceFetch can retrieve images effectively, it lacks sophisticated techniques to produce results efficiently. To this end, we propose SeekSuspect, a faster interactive suspect retrieval framework, which introduces several optimization algorithms to FaceFetch's framework. We train and test our system on a real-world dataset curated in collaboration with a metropolitan police department in India. Results reveal that SeekSuspect beats FaceFetch and can be employed by law enforcement agencies to retrieve suspects.

CCS CONCEPTS

• **Information systems** → *Image search.*

KEYWORDS

Suspect retrieval, user interaction, relevance feedback, active learning, optimisation techniques

ACM Reference Format:
Aayush Jain[1*], Meet Shah[1*], Suraj Pandey[1*], Mansi Agarwal[1*], Rajiv Ratn Shah[1], Yifang Yin[2], [1]Indraprastha Institute of Information Technology, Delhi, India, [2] National University of Singapore, Singapore, {aayush18122, shah18046, suraj18025, mansia, rajivratn}@iiitd.ac.in, yifang@comp.nus.edu. 2021. SeekSuspect : Retrieving Suspects from Criminal Datasets using Visual Memory. In *ACM Multimedia Asia (MMAsia '20), March 7–9, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3444685.3446252

1 INTRODUCTION

Despite the increase in security, the crime rate has been rising exponentially worldwide [9]. Most police departments maintain a crime dossier system that entails information regarding criminals like photographs and physical details [14]. Finding suspects by name is still possible from the database, but this method fails when we only have the informant's visual memory. Law enforcement agencies used to hire sketch artists, but they are limited in number, and the process is slow and inaccurate at times. To this end, several researchers have proposed facial retrieval systems. However, these require either input image [8] or concrete descriptions of the suspect [5, 7, 10], which are not always feasible.

FaceFetch [15] is a state-of-the-art framework that retrieves faces from an extensive database solely on visual memory. With slight modifications, FaceFetch could be extended as a suspect retrieval system. But, the time elapsed by FaceFetch during the retrieval is high and could hinder time-sensitive investigations. Moreover, the onlooker's memory could fade as time passes. Therefore, we propose SeekSuspect, a faster interactive suspect retrieval system that applies several optimization algorithms to the framework of FaceFetch to enhance its performance. Furthermore, we curated our dataset in collaboration with a metropolitan police department in India. Unlike benchmark datasets [4, 6], this dataset is noisy and an exemplary representative of real-world scenarios.

2 DATASET

With the aid of a metropolitan police department in India, we curated an extensive criminal based dataset[1]. Our dataset comprises $31,041$ data points. Each data point is associated with attributes describing the criminal's physical details like face shape, skin complexion etc. However, each instance does not have an associated image. We did not include such data points in our dataset. Furthermore, we omitted distorted images, which finally resulted in $20,304$ images and 153 attributes.

3 METHODOLOGY

SeekSuspect has a similar architecture as FaceFetch to retrieve the envisioned image. However, SeekSuspect employs several optimization algorithms to improve the existing FaceFetch framework. In the following subsections, we discuss the data preprocessing, introduce FaceFetch briefly, and present the proposed optimization techniques. SeekSuspect's architecture is depicted in Fig 1.

3.1 Data Preprocessing

Since our dataset is a real-world dataset, it is noisy and suffers from issues like missing values, class imbalance, etc. Hence, we employed sophisticated yet straightforward techniques to clean our data before feeding it into our system.

[*] These authors contributed equally to this work.

[1]Due to data privacy and confidentiality, we cannot release the dataset. The images in the paper are substitutes and not from the original dataset.

MMAsia '20, March 7–9, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8308-0/21/03.
https://doi.org/10.1145/3444685.3446252

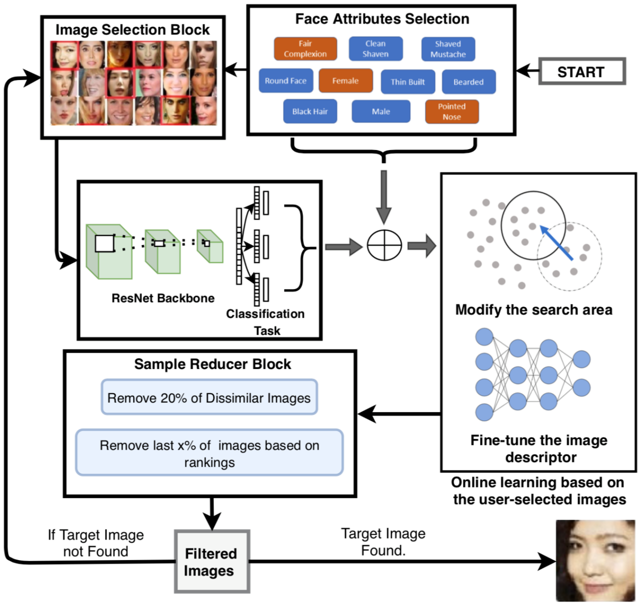

Figure 1: Architecture of SeekSuspect.

Images: We employed RetinaFace [2] to crop the face and remove unnecessary information from the image. Furthermore, some images in the dataset were rotated with 90, 180, or 270 degrees. So, we applied heuristics based on coordinates of the face detected by RetinaFace and used them to rotate the image correctly.

Attributes: We first removed those attributes that were not informative like *Speech*, *Language*, etc. To minimize class imbalance, we merged a few attributes. For example, *Fair Complexion* and *Very Fair Complexion* were integrated into a single class *Fair Complexion*. We further discarded attributes with very few instances, i.e., less than 10% of the dataset, such as *Leucoderma*.

After performing all these steps, we had 11,246 images associated with 28 categorical classes and one continuous class (*Height*).

3.2 FaceFetch

In the zeroth iteration of the process, FaceFetch asks the user to select some of the facial attributes they remember their envisioned image has. Based on these selected attributes $\mathcal{F}$, the user is presented with some images and asked to choose those that seem similar to the target image. The selected images $\mathcal{S}$ are then preprocessed [13] to obtain a semantic vector representation. Based on user selection, the search space $\mathcal{V}$ is modified [12], and the face descriptor network $\mathcal{D}$ [13] weights are updated. Updated descriptor $\mathcal{D}$ is used to obtain the new vector representation(s), and the images are re-ranked using cosine similarity distance metric δ. The system presents the user again with top $\mathcal{K}$ images, and the steps repeat until the user finds the envisioned image.

3.3 SeekSuspect

Unlike FaceFetch, we also reduce the search space after each iteration, which helps the process converge faster. We refer to the user-selected images at iteration i as *Similar Images* $\mathcal{S}_i$ and the rest as *Dissimilar Images* $\mathcal{S}'_i$. Let the search space at iteration i be $\mathcal{V}_i$. We compare all images in $\mathcal{V}_i$ with $\mathcal{S}_i$ and rank them using δ. $x\%$ least-ranked images are dropped every iteration, and x increases from 20, 30, 50 and so on as the iteration number i increases. We also remove 20% of the *Dissimilar Images* $\mathcal{S}'_i$ at every iteration. This threshold was large enough to reduce the search space, and small enough to not converge to a local minima. Furthermore, we constructed a classification model C that classifies facial attributes in each similar image $\mathcal{S}_i$. We trained C using Multi-Task Learning [1] with a Resnet50 model [3]. The intersection of these attributes is added to the selected attributes $\mathcal{F}$ to filter the sample space further.

Apart from search space reduction, we also introduced some techniques to reduce system latency. Unlike FaceFetch, we preprocessed and saved all the images beforehand. We also did not finetune our descriptor network $\mathcal{D}$ for fixed ten epochs every iteration. Instead, we employed Early Stopping [11]. We experimented with the batch size, and increasing it from 1 to 4 helped reduce the system latency further.

4 RESULTS

To compare the performance of SeekSuspect with FaceFetch, we randomly selected images one-by-one from the database. We then used both SeekSuspect and FaceFetch to retrieve the same image based on visual memory (i.e., same facial attributes). We recorded the total search time and the total number of iterations the systems took to search for the target image.

Table 1 shows that, on average, SeekSuspect is six times faster than FaceFetch and can retrieve results in lesser search iterations than FaceFetch, thereby proving to be a more efficient and effective suspect retrieval system.

System	**Average Time Taken (s)**	**Average Number of Iterations**
FaceFetch [15]	193.16	5.36
SeekSuspect	**30.04**	**4.00**

Table 1: Comparison with state-of-the-art.

We also calculated precision as the number of times SeekSuspect was able to find the envisioned face in top-($\mathcal{K}$) images in N number of iterations. Table 2 reports the precision for $\mathcal{K}$ = 1, 5, 10 and 15 for N = 8. As expected, the performance improves as $\mathcal{K}$'s value increases. However, with an increase in the suspect list's size ($\mathcal{K}$), investigations might run longer, proving costly in time-sensitive scenarios.

	$\mathcal{K}$ = 1	$\mathcal{K}$ = 5	$\mathcal{K}$ = 10	$\mathcal{K}$ = 15
Precision	0.23	0.35	0.47	0.64

Table 2: Effect of $\mathcal{K}$ on SeekSuspect's performance.

5 CONCLUSION AND FUTURE WORK

This paper presents SeekSuspect, an interactive suspect retrieval system capable of procuring suspects solely based on the informant's visual memory. SeekSuspect introduces several optimization techniques to the FaceFetch framework, resulting in a more efficient and effective system. However, like FaceFetch, one of SeekSuspect's drawbacks is that it can only retrieve those suspects whose information is present in the criminal dossier system. We plan to work towards this avenue in the future.

ACKNOWLEDGEMENTS

Rajiv Ratn Shah is partly supported by the Infosys Center for AI and the Center of Design and New Media at IIIT Delhi.

REFERENCES

[1] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.

[2] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. 2019. Retinaface: Single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641* (2019).

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[4] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. 2008. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments.

[5] Mohd Aamir Khan and Anand Singh Jalal. 2020. A framework for suspect face retrieval using linguistic descriptions. *Expert Systems with Applications* 141 (2020), 112925.

[6] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*. 3730–3738.

[7] Osaid Rehman Nasir, Shailesh Kumar Jha, Manraj Singh Grover, Yi Yu, Ajit Kumar, and Rajiv Ratn Shah. 2019. Text2FaceGAN: Face Generation from Fine Grained Textual Descriptions. In *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*. IEEE, 58–67.

[8] Jyoti Nautiyal, Shivali Gahlot, and Pawan Kumar Mishra. 2013. An automated technique for criminal face identification using biometric approach. In *Proceedings of the Conference on Advances in Communication and Control Systems-2013*. Atlantis Press.

[9] Clive Norris, Mike McCahill, and David Wood. 2004. The growth of CCTV: a global perspective on the international diffusion of video surveillance in publicly accessible space. *Surveillance & Society* 2, 2/3 (2004).

[10] Sharan Pai, Nikhil Sachdeva, Rajiv Shah, and Roger Zimmermann. 2019. User Input Based Style Transfer While Retaining Facial Attributes. In *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*. IEEE, 68–76.

[11] Lutz Prechelt. 1998. Early stopping-but when? In *Neural Networks: Tricks of the trade*. Springer, 55–69.

[12] Joseph Rocchio. 1971. Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing* (1971), 313–323.

[13] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.

[14] Rajiv Shah and Roger Zimmermann. 2017. *Multimodal analysis of user-generated multimedia content*. Springer.

[15] Harsh Shrivastava, Rama Krishna PVNS, Karmanya Aggarwal, Meghna P Ayyar, Yifang Yin, Rajiv Ratn Shah, and Roger Zimmermann. 2019. FaceFetch: An Efficient and Scalable Face Retrieval System that uses your Visual Memory. In *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*. IEEE, 338–347.