Bauhaus-Universität Weimar

Fakultät Bauingenieurwesen

IMAGE DEDUPLICATION (IMDEDU)

Olubunmi Emmanuel Ogunleye

Selva Ganapathy Elangovan

Special Project Presentation

Name Of Advisor: Prof. Björn Rüffer

07-03-2025

Project Presentation Outline

Introduction / Motivation

How Do We Detect Image Duplicates?

Significance #1

Eliminate Multiple Versions of Same Image

Significance #2

Efficient Usage of Data Storage Space

Significance #3

Reduce Search Complexity in Information Retrieval

Aim and Objectives

Aim is to Implement an Algorithm that Detects Near Duplicates.

01

Implement a way to gather collection of images from data storage

02

Define metric to measure similarities between images

03

Implement a technique to cluster similar images

04

Determine the best version of a group of similar images

Literature Review

Theory 01

Hash Functions

Mathematical algorithm that transforms data into fixed strings of bit called hash codes.

(Ref: Stallings, 2017)

Theory 02

Cryptographic Hashes

Unique and consistent output for any given input. Minor change result in significantly different hash value. Collision Resistant. Examples: MD5, SHA-256 (Ref: Schneier, 2017)

Theory 03

Perceptual Hashes

Finding similarities between Images. Collisions can occur. Produces similar hash values for similar images.

(Sharma, 2014)

Theory 04

Similarity Distance Measures

Used to cluster similar images. Examples Euclidean, Hamming, Manhattan etc.

Literature Review Cont'D

Types of Perceptual Hashes (Ref: Viies, 2015)

Average Hash

Averages the pixel values. Computationally Efficient

Median Hash

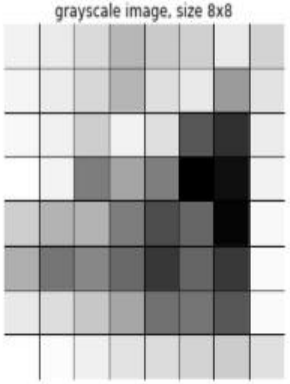
Uses Median Pixel Values. Robust to Certain Image Modification

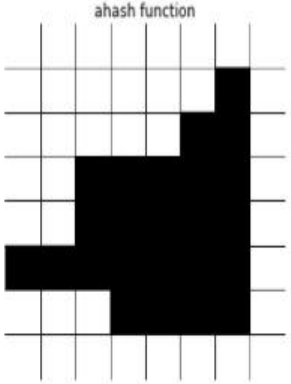
Perceptual Hash

Uses Discrete Cosine Transform. Robust to scaling and Rotations

Wavelet Hash

Uses Discrete Wavelet Transformation. Also extract hash values from low frequencies.





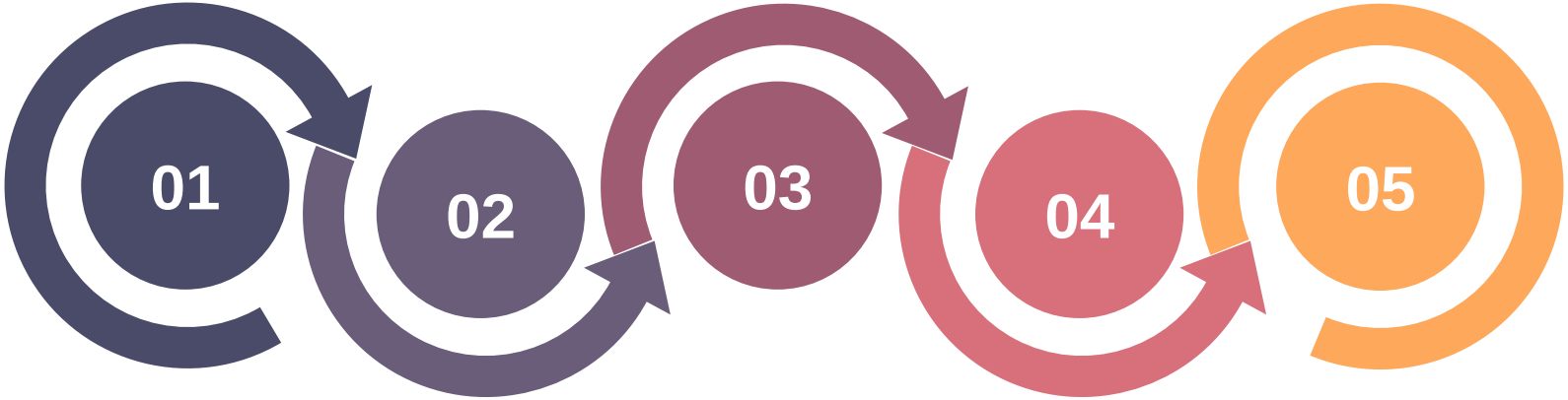
Literature Review Cont'D

Average Hashing (Ref: Viies, 2015)

Step 01
Resize image into a fixed dimension 8x8 pixels

Step 03
Determine the Average Pixel Value

Step 05
Convert Binary String to Hexadecimal Strings



Step 02
Convert resized image to Grayscale

Step 04
Compute Binary Hash Value by Thresholding with Average Pixel Value. Result is a Binary String

Literature Review Cont'D

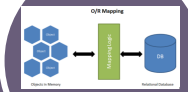
Hamming Distance Metric

Comparing two binary vectors bit by bit.

Differing bits are assigned value of 1

Similar bits are assigned value of 0

(Ref: Hamming, 1950)



SQL Alchemy

Object Relational Mapper (ORM)

Data Storage

Flexibility to Extend

(Ref: Bayer, 2025)

Streamlit

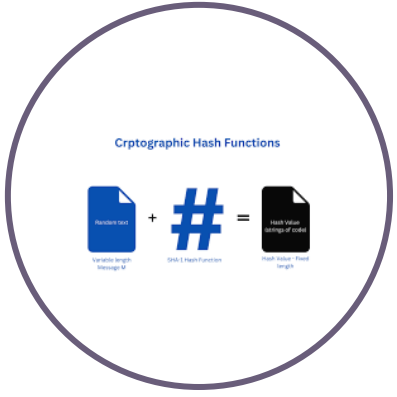
Open Source Python Frameworks for Interactive Web Applications

(Ref: Streamlit Community, 2025)

Methodology

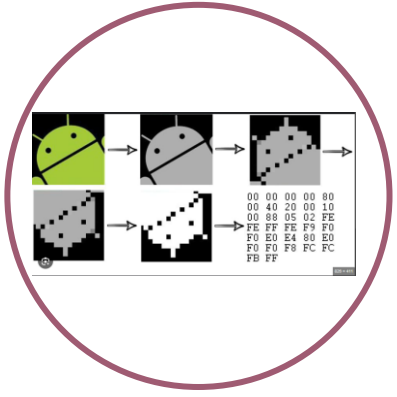
Load Images and Image Properties from Data Storage

(Local Drive or Hard Drive)



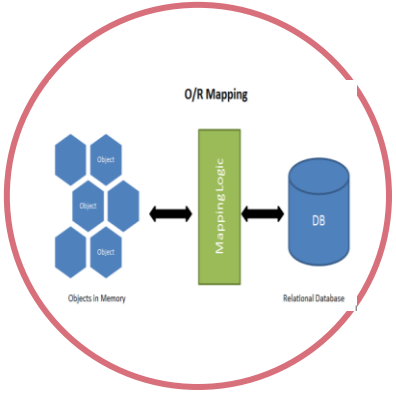
Compute Cryptographic Hash Values as Key to Store Image Metadata in SQL Database

SHA-256 was used



Compute Perceptual Hashes. Involves creating 3 instances along side original.

Average hash was used.

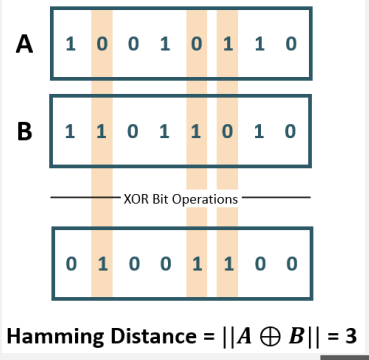


SQL Alchemy to store Image MetaData and Perceptual Hashes into DB. Cryptographic Hash used as Key

Methodology Cont'D

Distance Metrics

- Using Hamming Distance.



Similarity Threshold

- Implementing a slider to allow user defined similarity threshold.
- Limits of the threshold are based on max. and min. value of the minimum pairwise hamming distance.

Image Ranking

- From Similar Image Cluster, a ranking criteria is implemented to choose best fit.
- This ranking criteria is based on image with best resolution.

Visualization

- Streamlit is used to create web based UI.
- This helps users interact with the built algorithm.
- Takes in user input and returns output.

Result and Discussion

Cryptographic Hash Result

64-character long hexadecimal string. Proved effective for storing Images MetaData.

Perceptual Hash Result

16-character long hexadecimal string. Proved effective for detecting similarities between images.

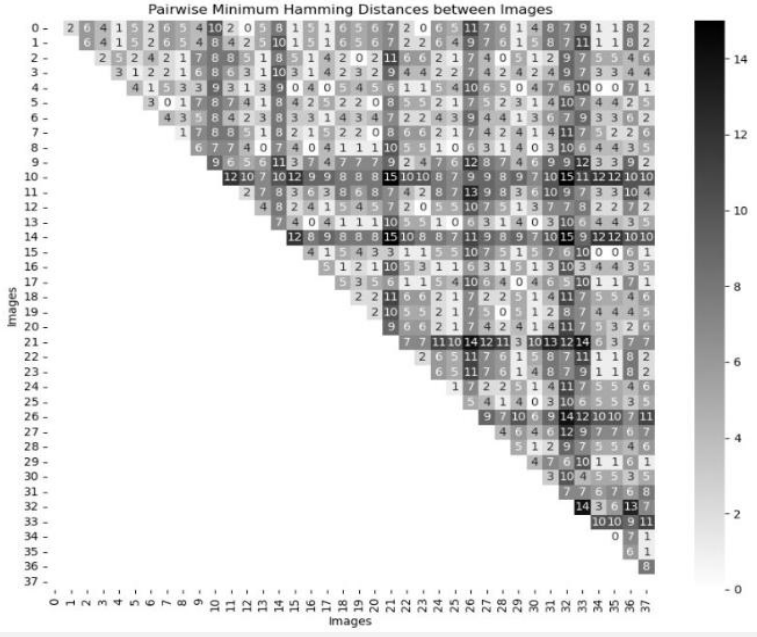
Similarity Measure

Hamming Distance was effective in measuring similarity between images. Similarity Matrix was plotted.

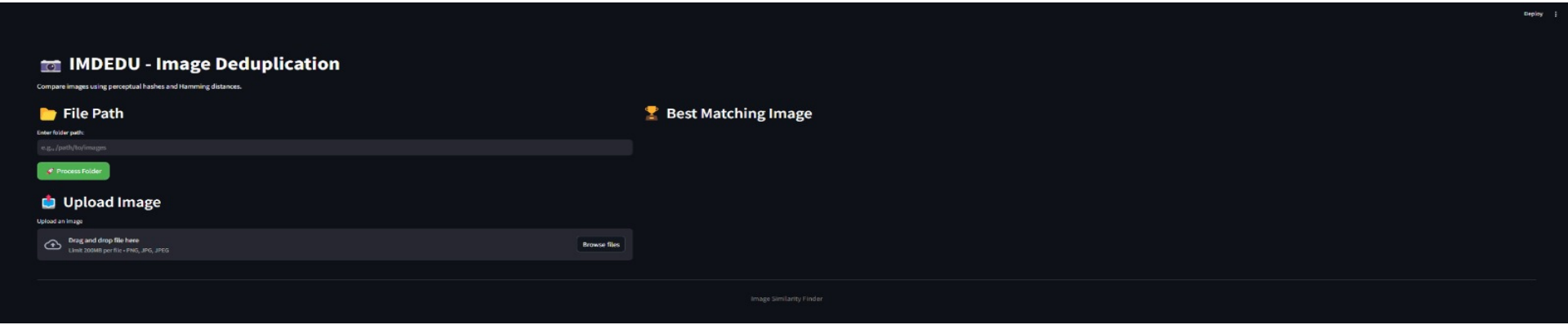
Streamlit for Visualization

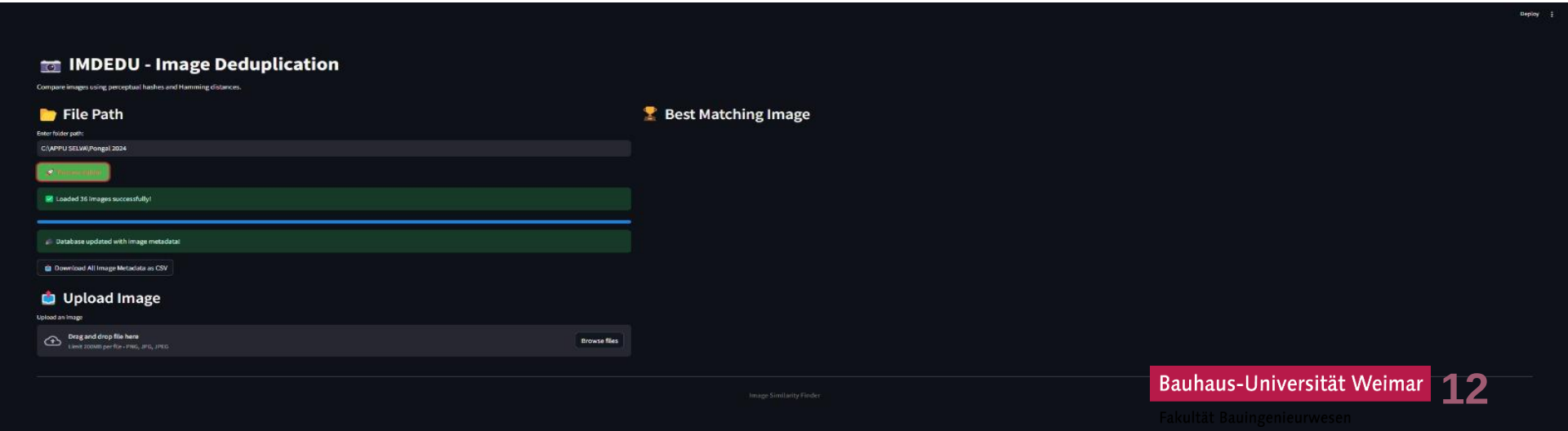
Performed effectively to provide efficient user interaction with algorithm.


```
{
  '1f1a20e54273cf20d5342de29e7591b829cede977040303bc8cd73aac871212d': {
      'Perceptual_Hash': '3f3b380d1e3c6400',
      'Perceptual_Hash_rotation_15degrees_Clockwise': '1e7efefffff7f7e70',
      'Perceptual_Hash_rotation_15degrees_CounterClockwise': '787f7ffffffe7e0e',
      'Perceptual_Hash_Shear_Low_Difference': '3f3bba3e1efefe00'
  },
  '0046be088fdafdb8d921d937ce119730585b21d8f26a974d4000d6d26707Sfb44': {
      'Perceptual_Hash': '0040f8fcfeffffff',
      'Perceptual_Hash_rotation_15degrees_Clockwise': '0e7efefffff7f7e78',
      'Perceptual_Hash_rotation_15degrees_CounterClockwise': '787e7ffffffe7e1e',
      'Perceptual_Hash_Shear_Low_Difference': '00fefefefefefe80'
  },
```

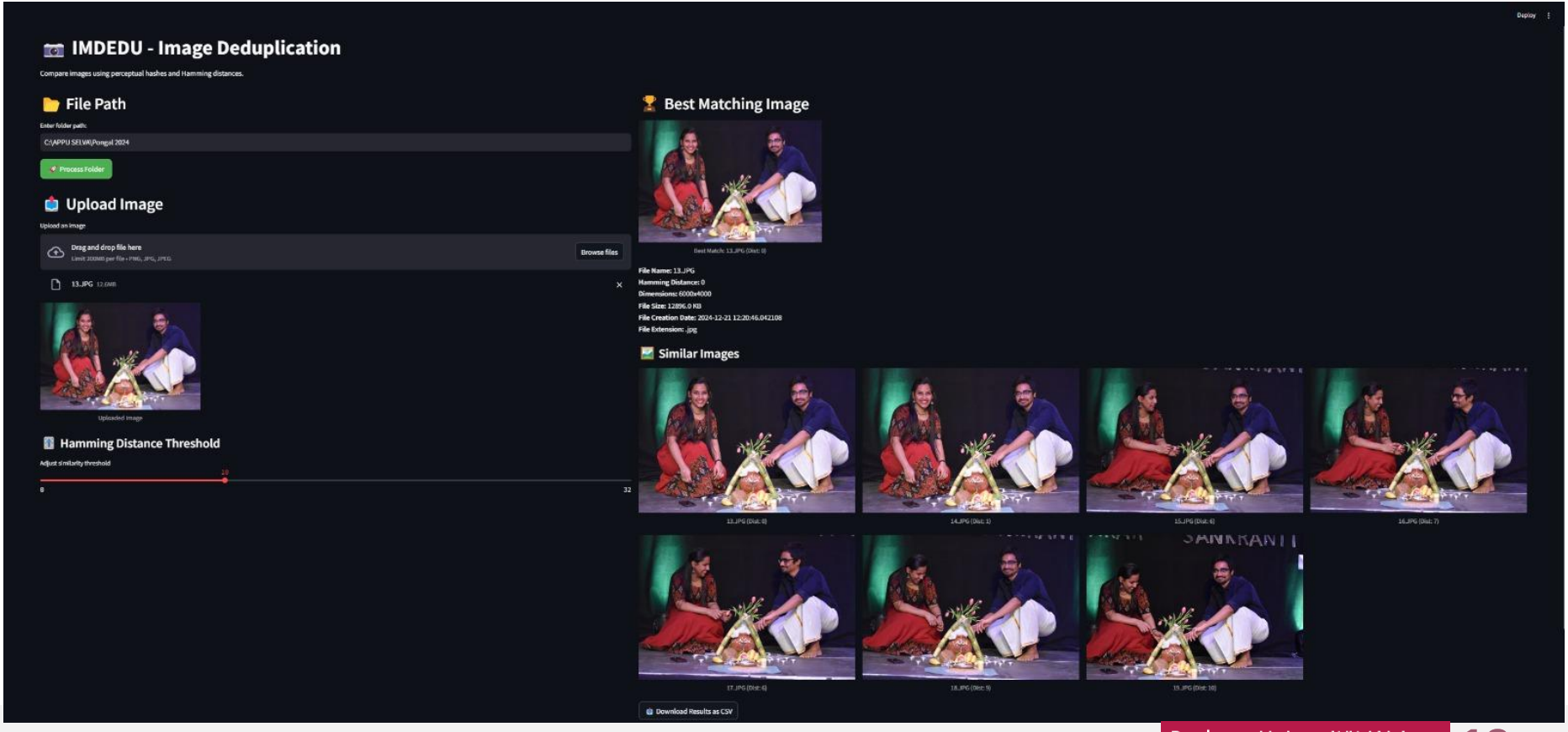



Result and Discussion Cont'D





Result and Discussion Cont'D



Conclusions / Findings

Cryptographic Hashes efficient for storing and retrieval of image information

Perceptual Hash effective for identifying Similarities

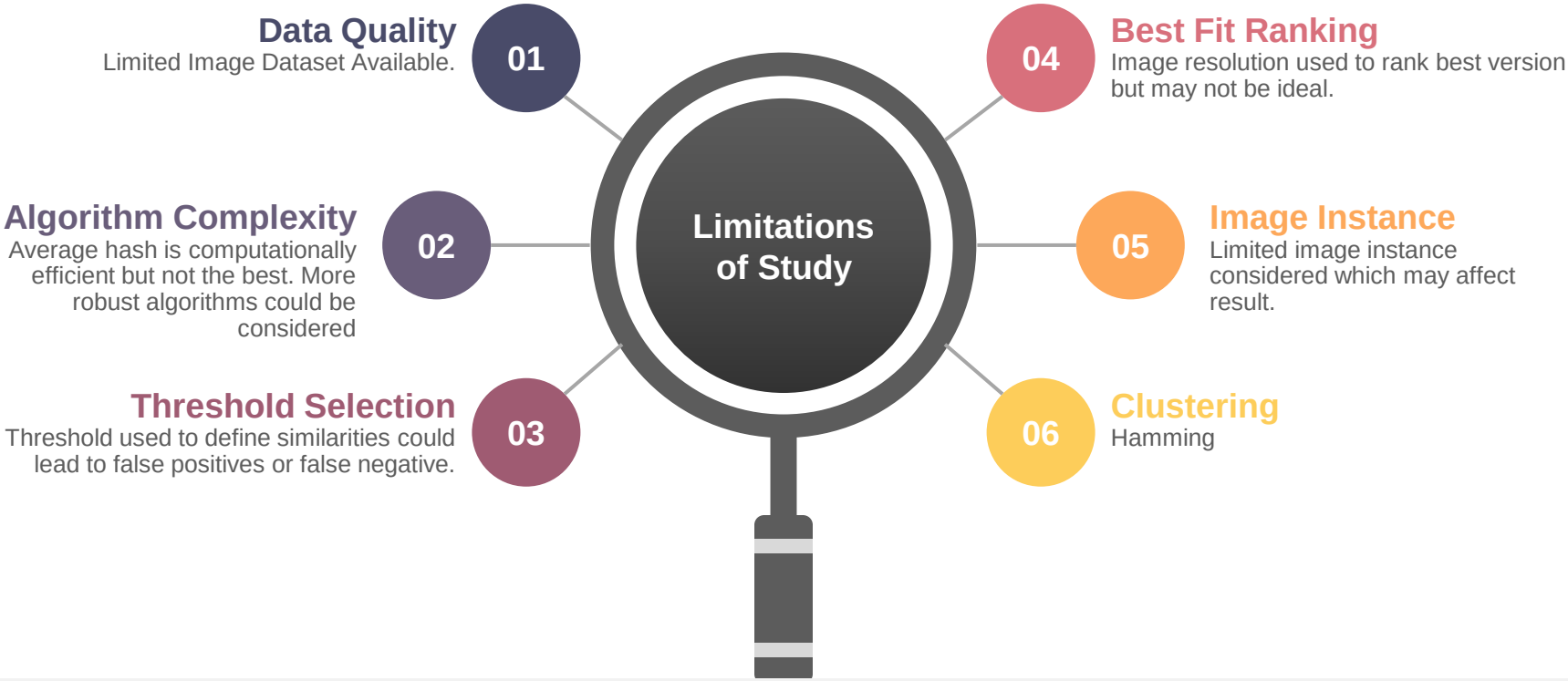
Hamming Distance proved effective for measuring similarities

SQL Alchemy effective for ORM and Image MetaData Storage

Streamlit provided user friendly GUI

Proposed Hash technique effective for Image Deduplication

Limitations of Study



Recommendations

Recommendation 1

Research should be expanded to include more sophisticated algorithms such as pHash to improve accuracy.

Recommendation 2

Exploring modern methods like machine learning-based hashing techniques could offer greater flexibility and adaptability to identify similarities.

Recommendation 3

Future work could involve testing larger and more diverse datasets including images from sources such as social media, scientific data and websites.

References

1. Aradhana and D. S. M. Ghosh. Review Paper on Secure Hash Algorithm With Its Variants. nternational Journal of Technical Innovation in ModernEngineering & Science, 3(5):01–07, Nov. 2021.

2. M. Bayer. SQLAlchemy - The Database Toolkit for Python, 2025. URLhttps://www.sqlalchemy.org/. Version 2.0, Accessed: 2025-03-04.

3. B. Schneier. Applied Cryptography: Protocols, Algorithms and Source Code inC. Wiley, 2017.

4. R. W. Hamming. Error Detecting and Error Correcting Codes. Bell SystemTechnical Journal, 29(2):147–160, 1950.

5. S. Sharma. Distance distributions and runtime analysis of perceptual hashingalgorithms. Journal of Visual Communication and Image Representation,104:104310, 2024. doi: 10.1016/j.jvcir.2024.104310.

6. W. Stallings. Cryptography and Network Security: Principles and Practice.Pearson, 7th Global Edition edition, 2017.

7. V. Viies. Possible Application of Perceptual Image Hashing. Tallinn Universityof Technology, Faculty of Information Technology Department of ComputerEngineering, Master Thesis, 2015.

8. Streamlit Community. Streamlit: Turn Python Scripts into Beautiful WebApps, 2025. URL https://docs.streamlit.io/get-started. Version1.42.0, Accessed: 2025-02-05.

QUESTIONS

THANK YOU FOR LISTENING!

Bauhaus-Universität Weimar
Fakultät Bauingenieurwesen