Comparison of Neural Networks for Reverse Image Searching

Presented by Andy Zhu and EstherNanjala Wekesa

- **Rapid Rise Of Machine Learning**

Chat GPT have taken a growing role in our lives

- **Issue with Algorithms**

(Lew et. al., 2006)

- **Issue with Algorithms**

  - Algorithms are a black box

(Lew et. al., 2006)

Issue with Algorithms

- Algorithms are a black box
- **Unclear if defined architecture is utilized as expected**

(Lew et. al., 2006)

Issue with Algorithms

Lack of diversity in data can unintentionally lead to disenfranchisement

(Lew et. al., 2006)

Issue with Algorithms

Lack of diversity in data can unintentionally lead to disenfranchisement

Important to input known data and observe how model behaves

(Lew et. al., 2006)

- **Current Popular Applications**

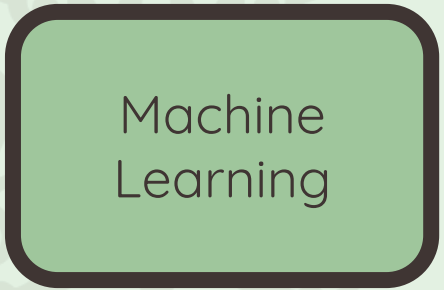


Figure 1

- **Current Popular Applications**

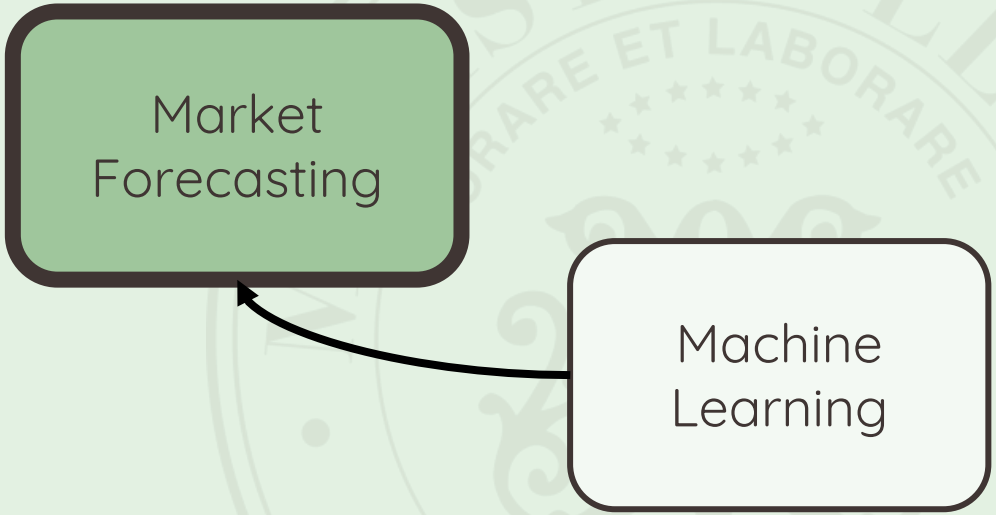


Figure 1

- **Current Popular Applications**

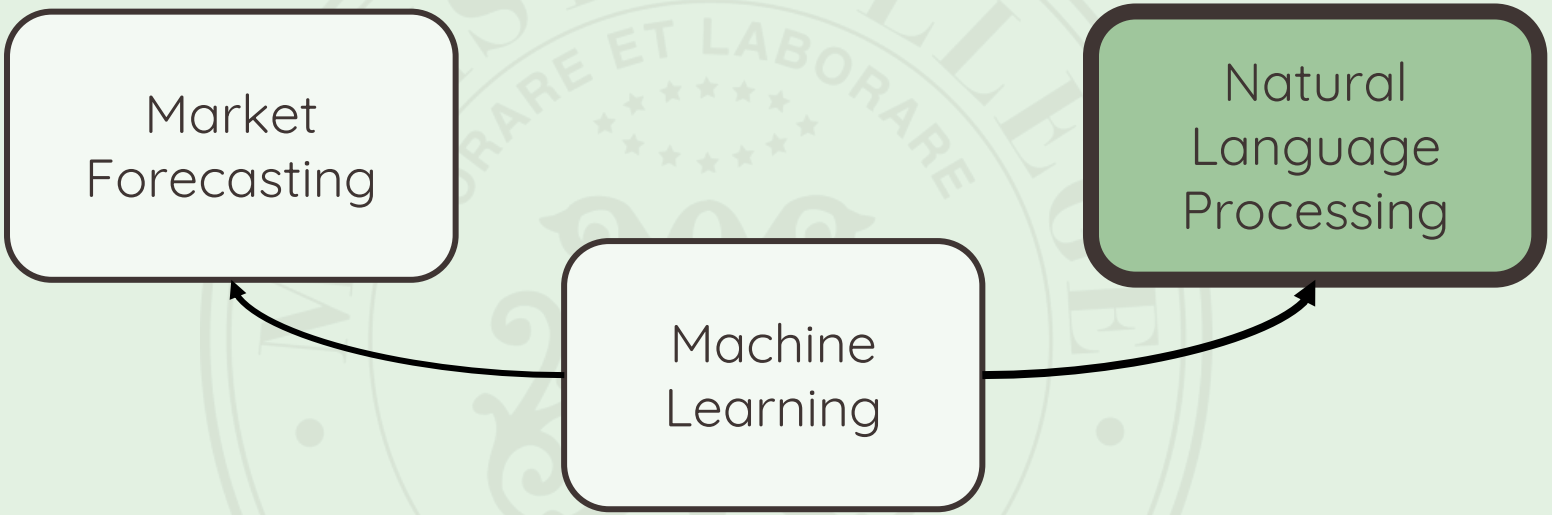


Figure 1

- **Current Popular Applications**

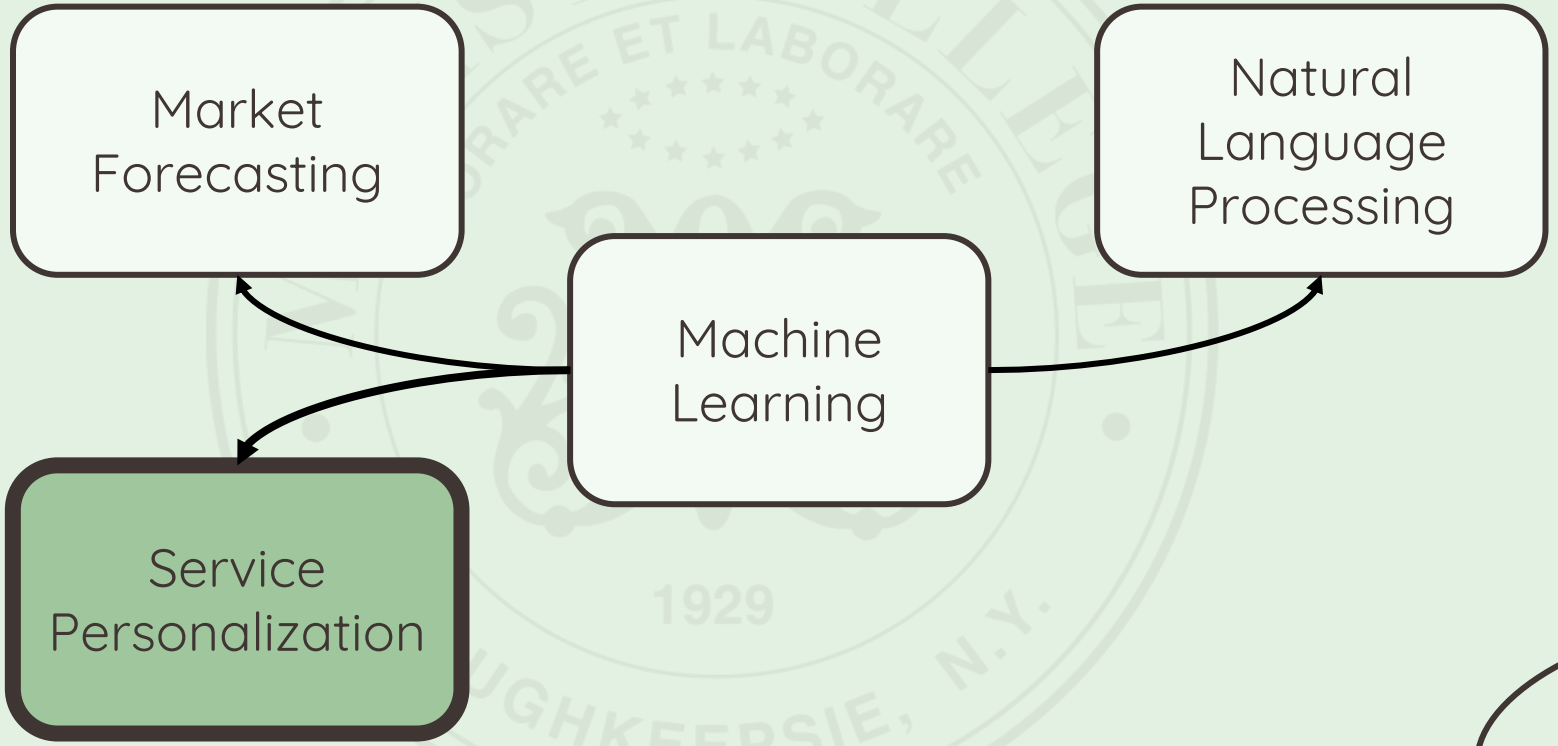
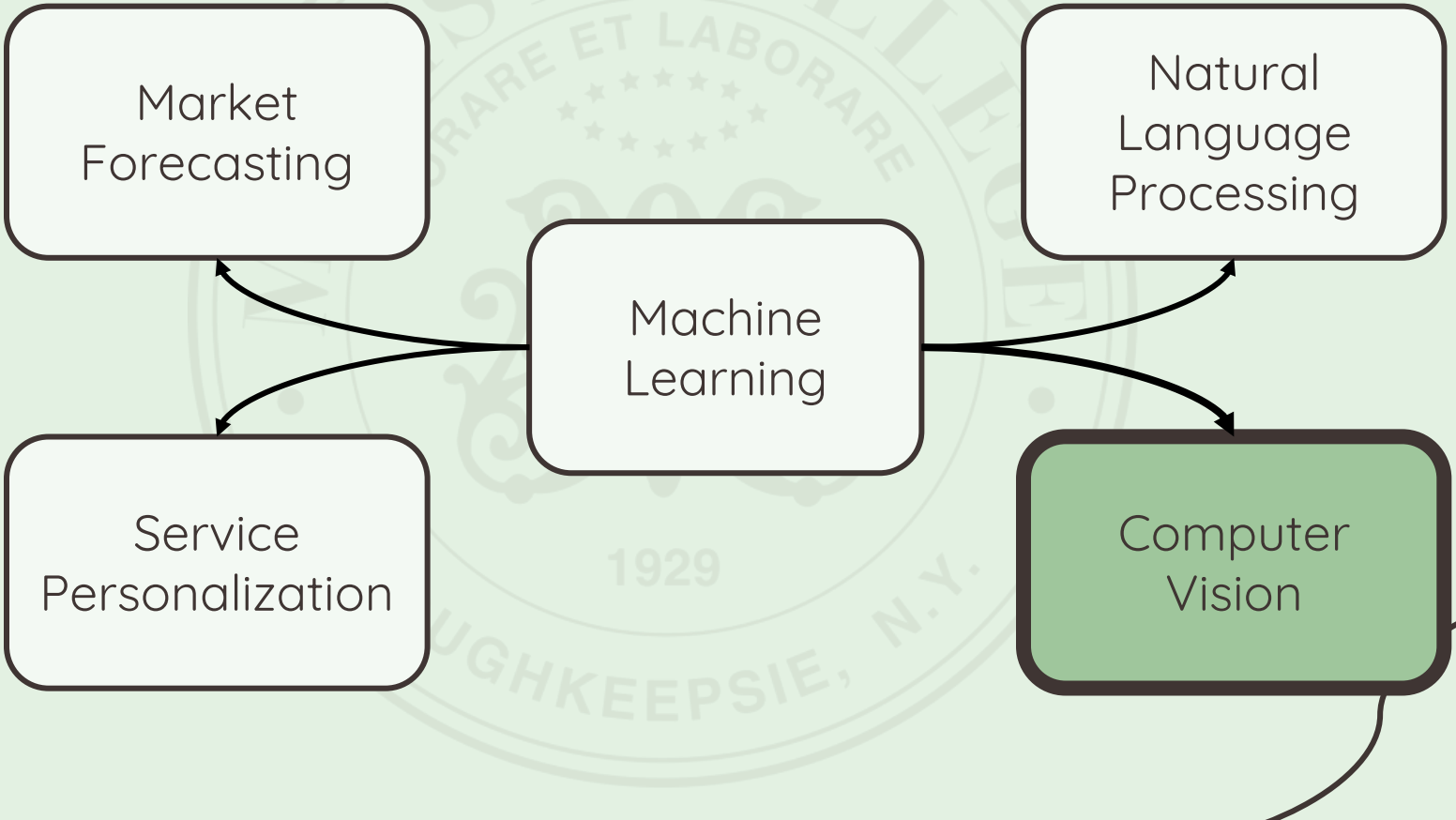
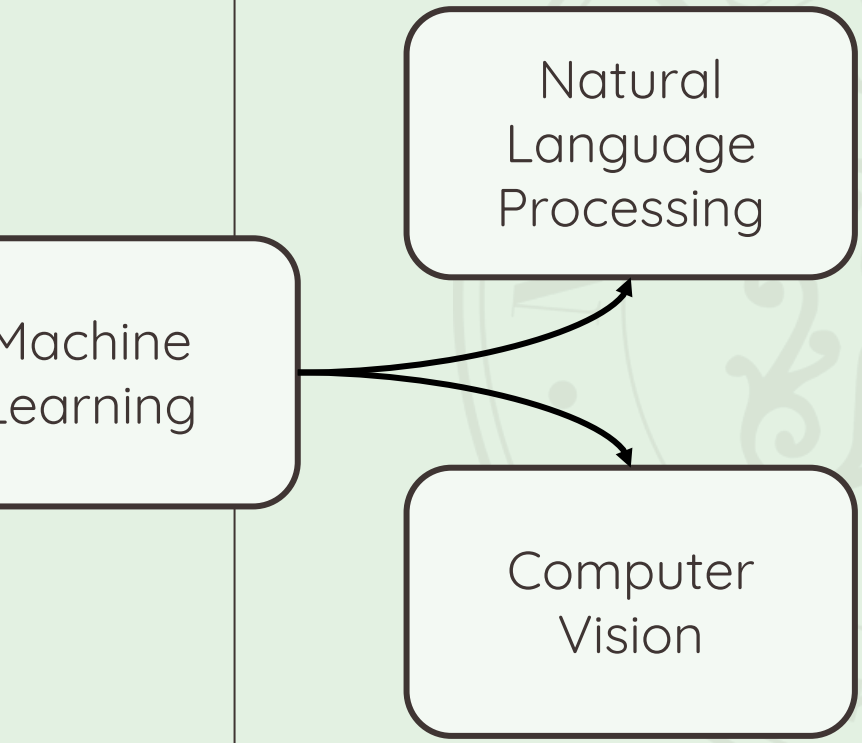


Figure 1

- **Current Popular Applications**



- **Current Popular Applications**



- **Current Popular Applications**

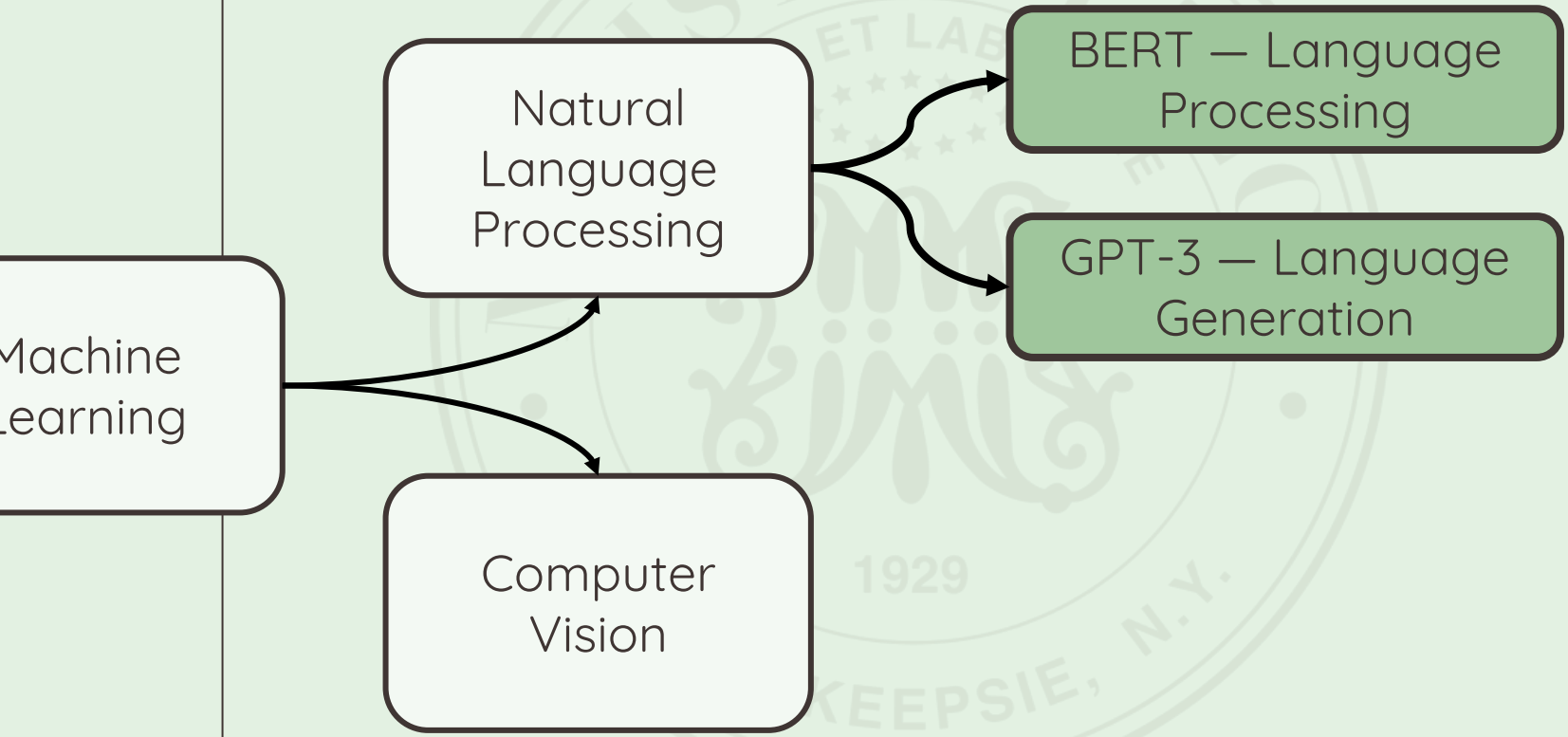
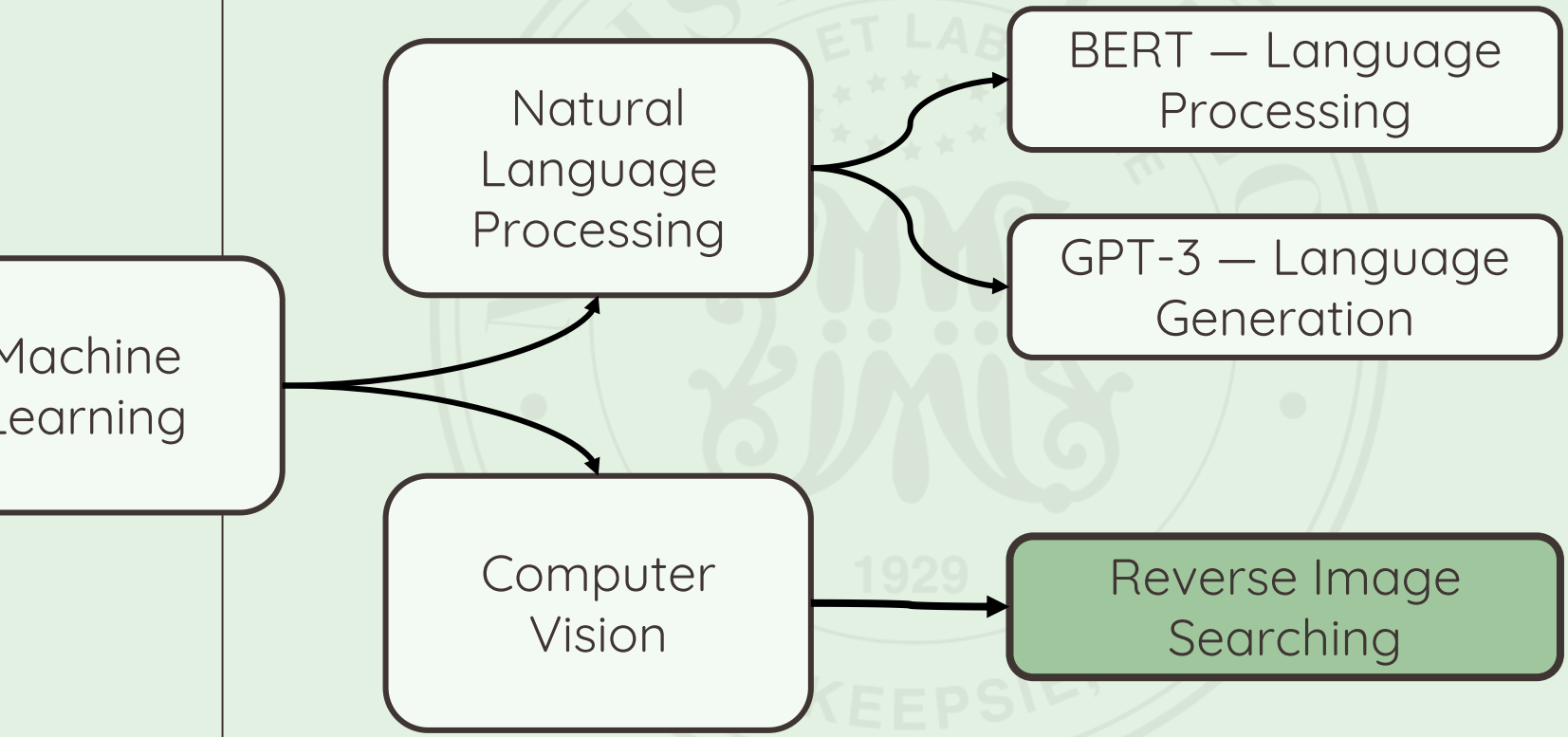


Figure 1

- **Current Popular Applications**



Reverse Image Search Algorithms

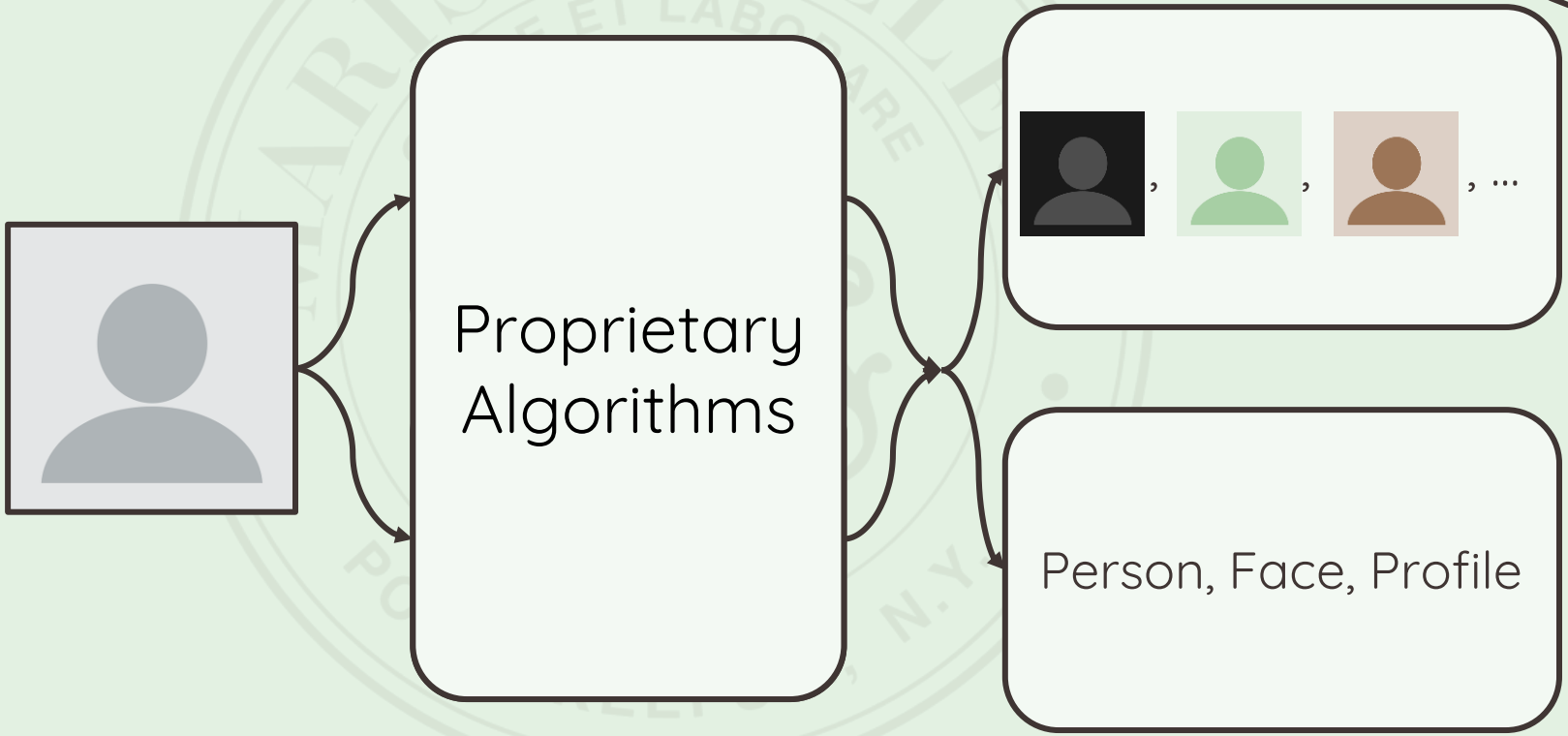


Figure 2

(Lew et. al., 2006)

Reverse Image Search Algorithms

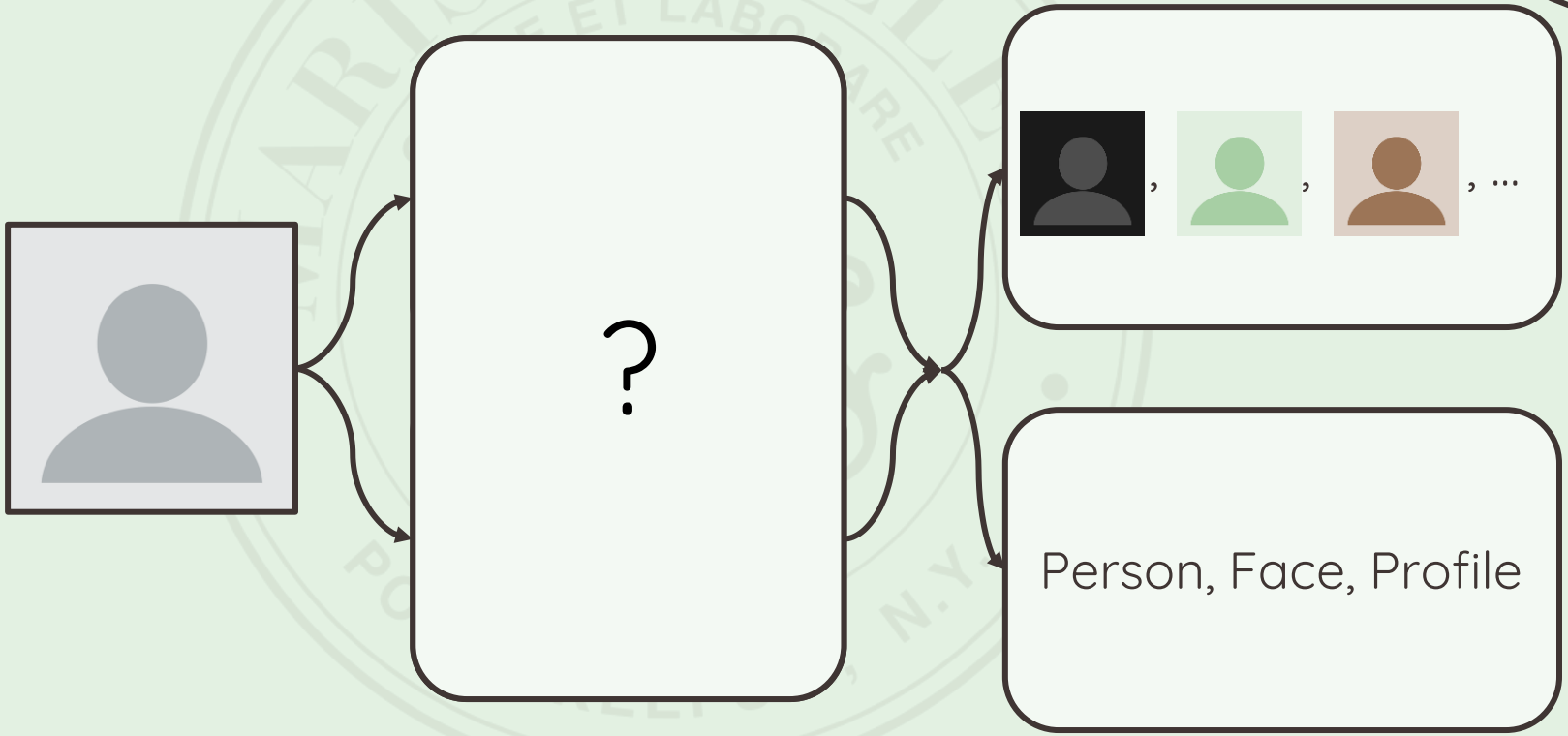


Figure 2

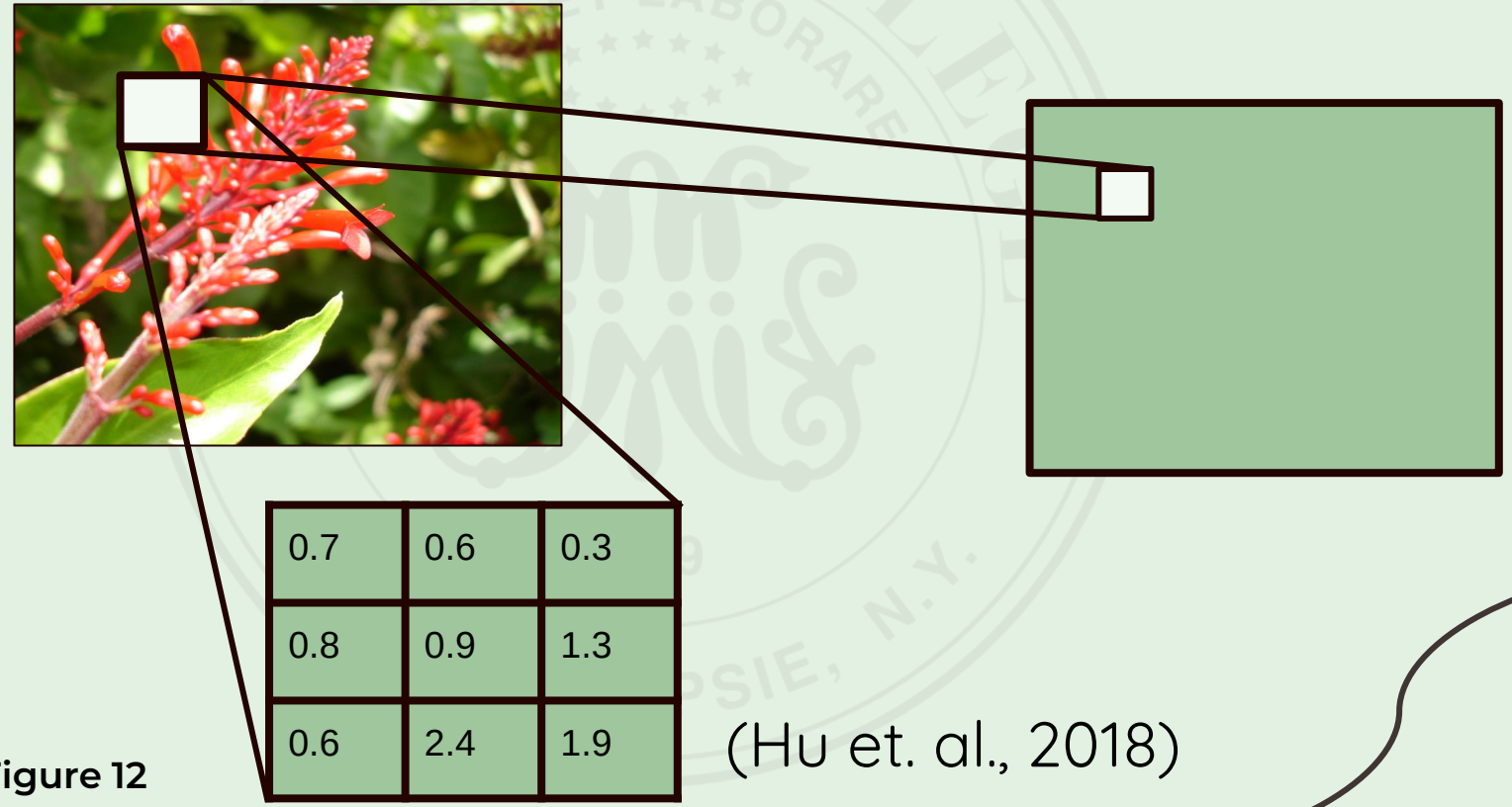
(Lew et. al., 2006)

Investigation of Algorithms

Companies not willing to explicitly disclose utilized algorithms

Compiled general algorithms utilized by investigating published articles and patents

- **Convolutional Neural Networks**



(Hu et. al., 2018)

Figure 12

- **Convolutional Neural Networks**

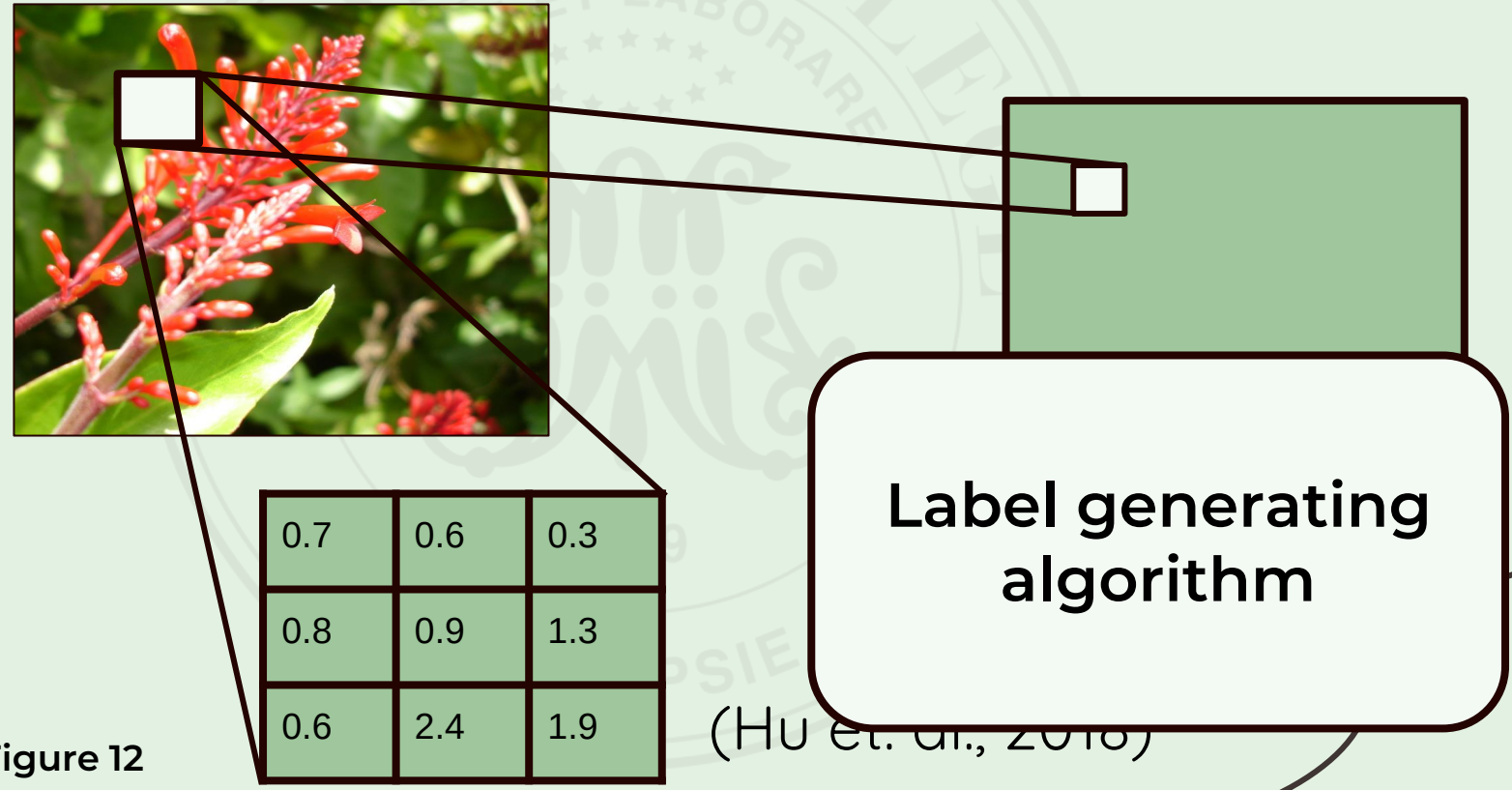
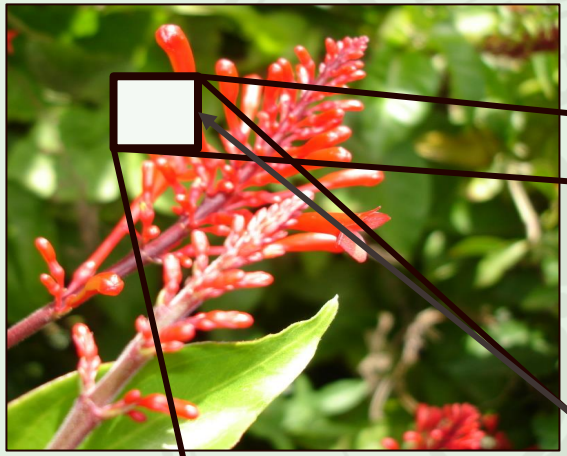


Figure 12

(Hu et. al., 2018)

- **Convolutional Neural Networks**



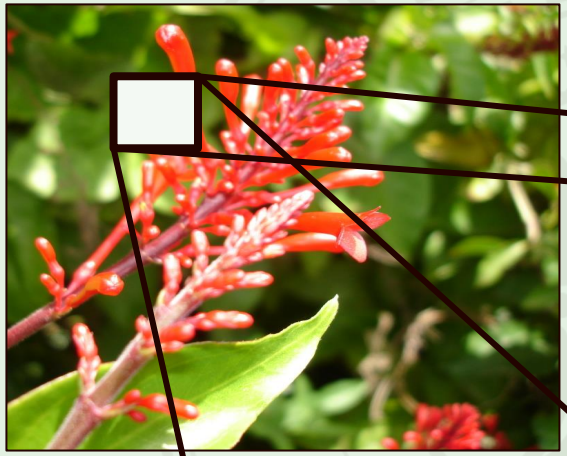
0.7	0.6	0.3
0.8	0.9	1.3
0.6	2.4	1.9

Passes convolution over entire image

(Hu et. al., 2018)

Figure 12

- **Convolutional Neural Networks**



0.7	0.6	0.3
0.8	0.9	1.3
0.6	2.4	1.9

Convolutional layers followed by pooling layers to compress image

Figure 12

(Hu et. al., 2018)

- **Convolutional Neural Networks**

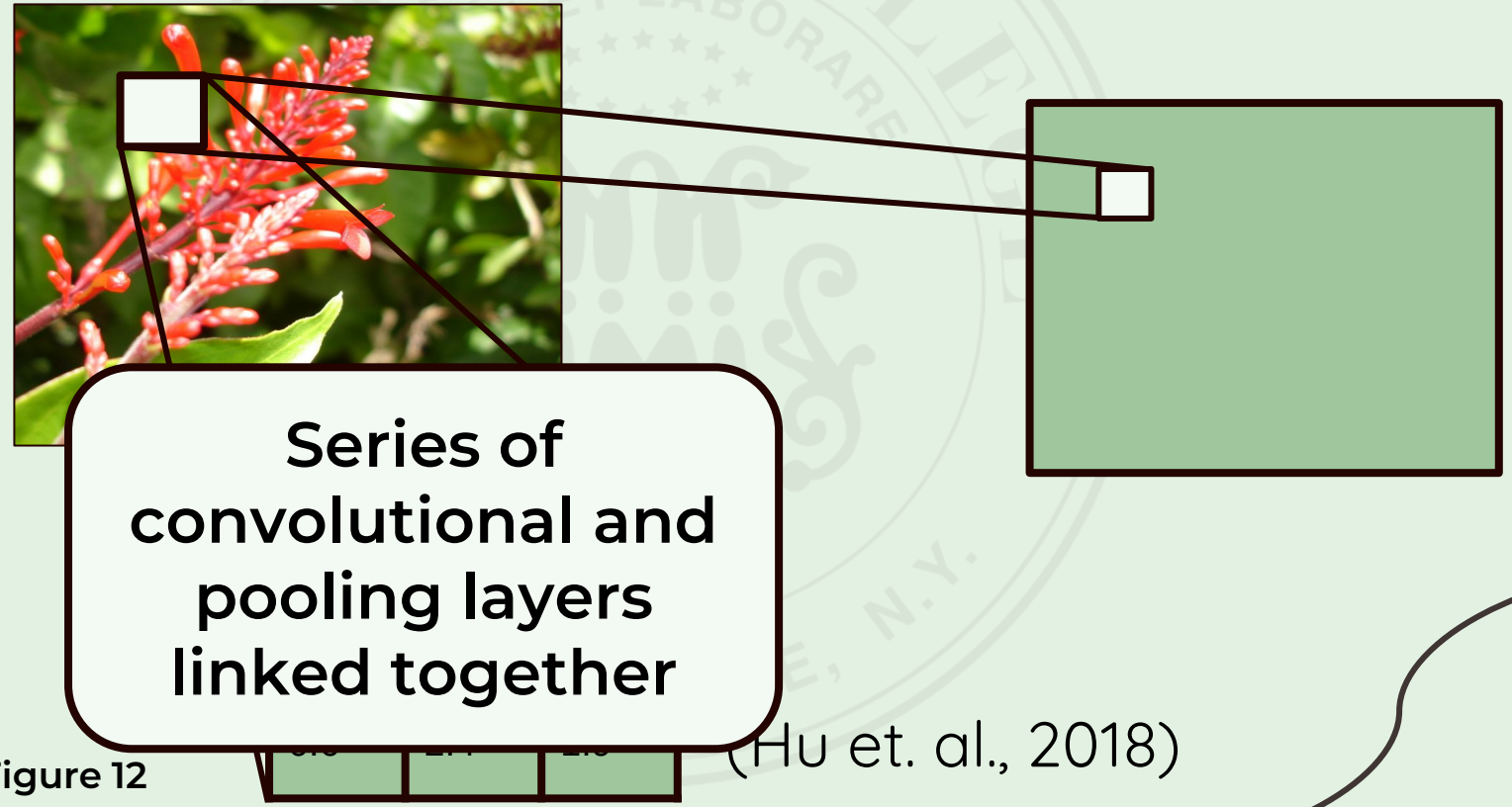


Figure 12 (Hu et. al., 2018)

- **Hashing Function** (Koul, et. al., 2009)

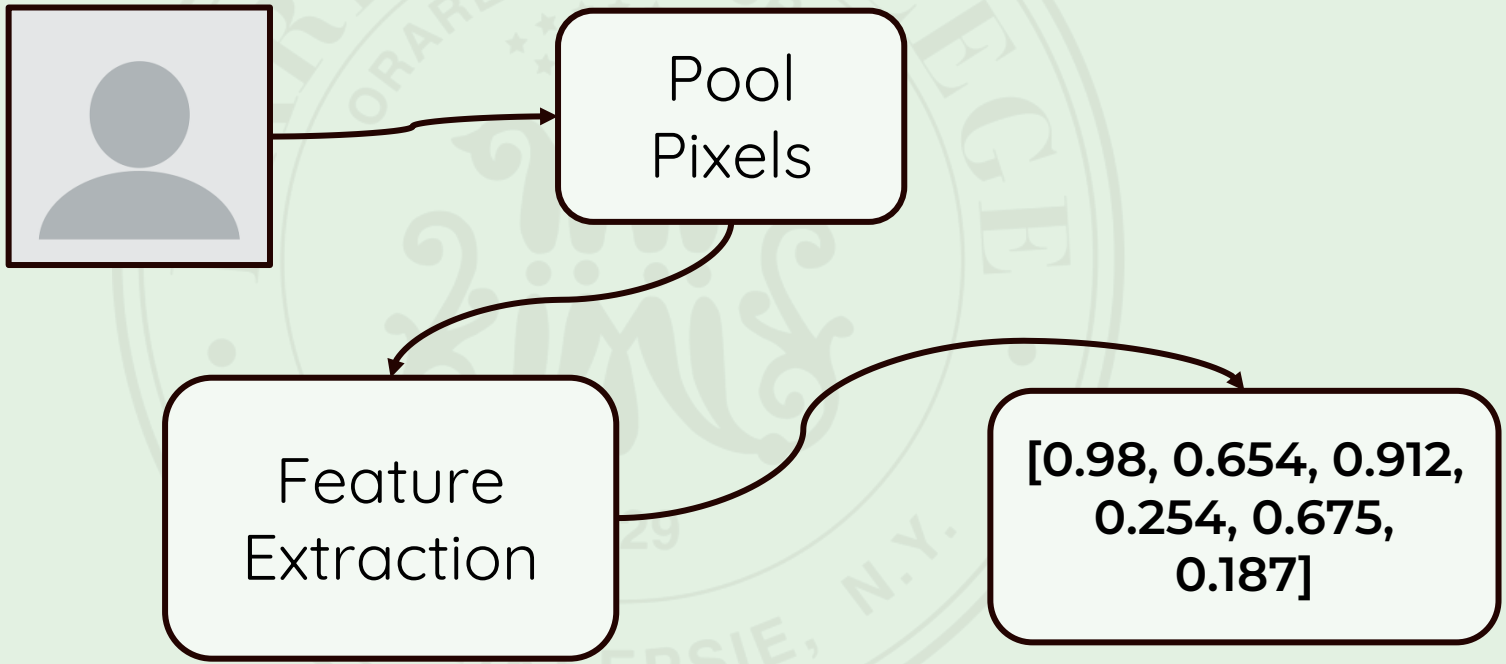


Figure 9

- **Perceptual Hashing** (Koul, et. al., 2009)

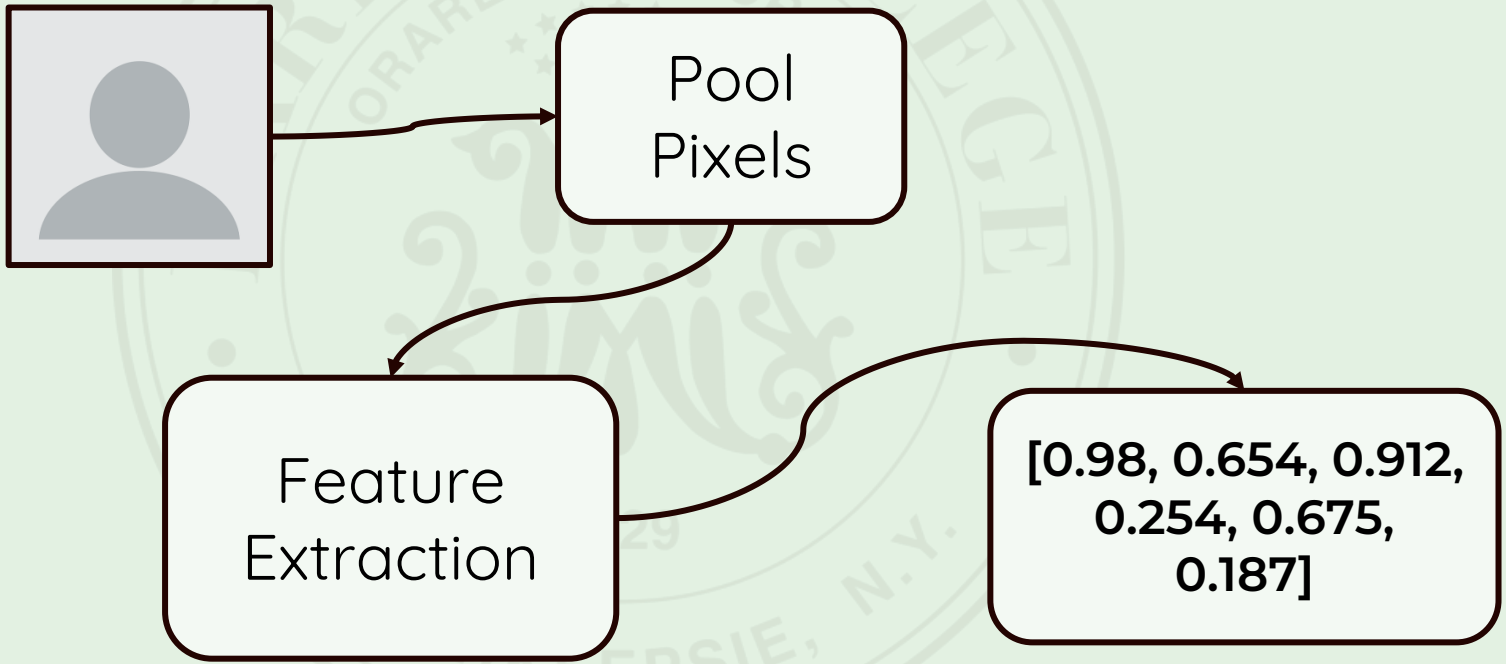


Figure 9

- **Perceptual Hashing** (Koul, et. al., 2009)

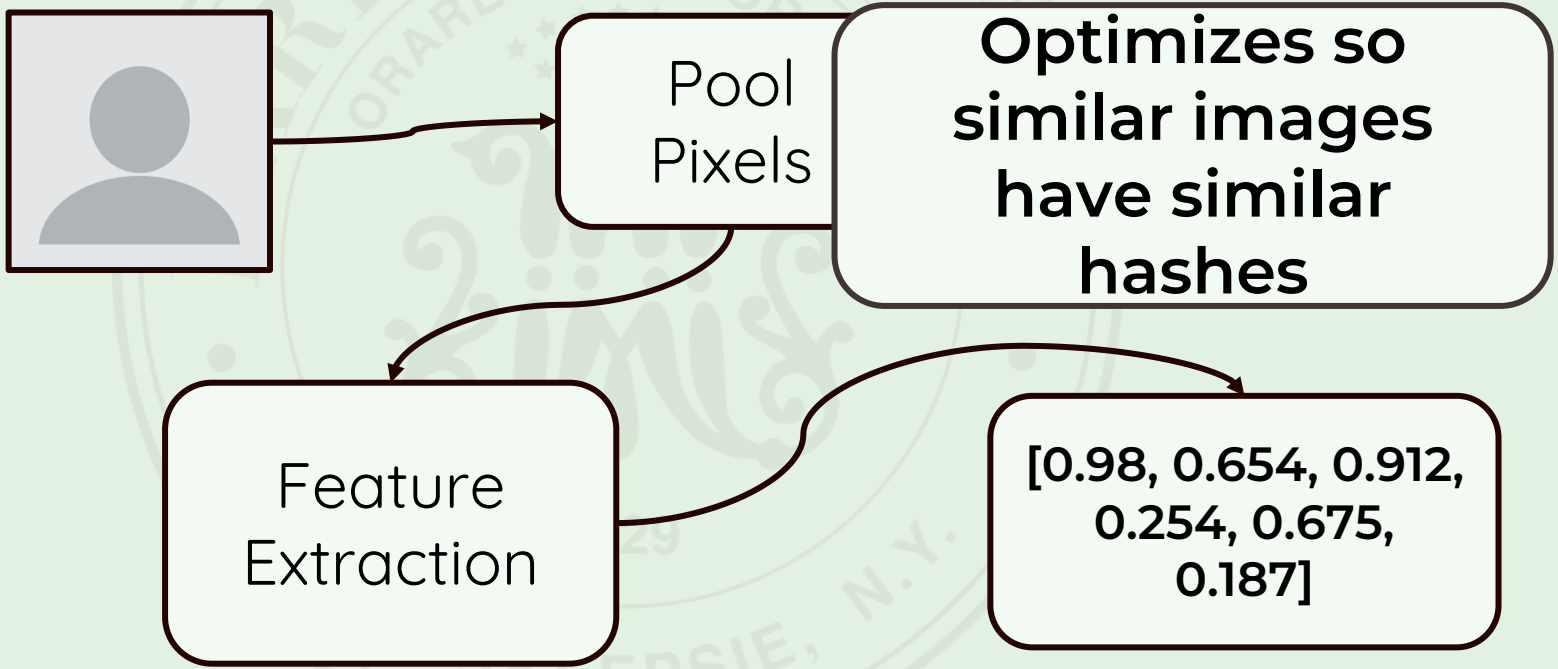


Figure 9

- **Perceptual Hashing** (Koul, et. al., 2009)

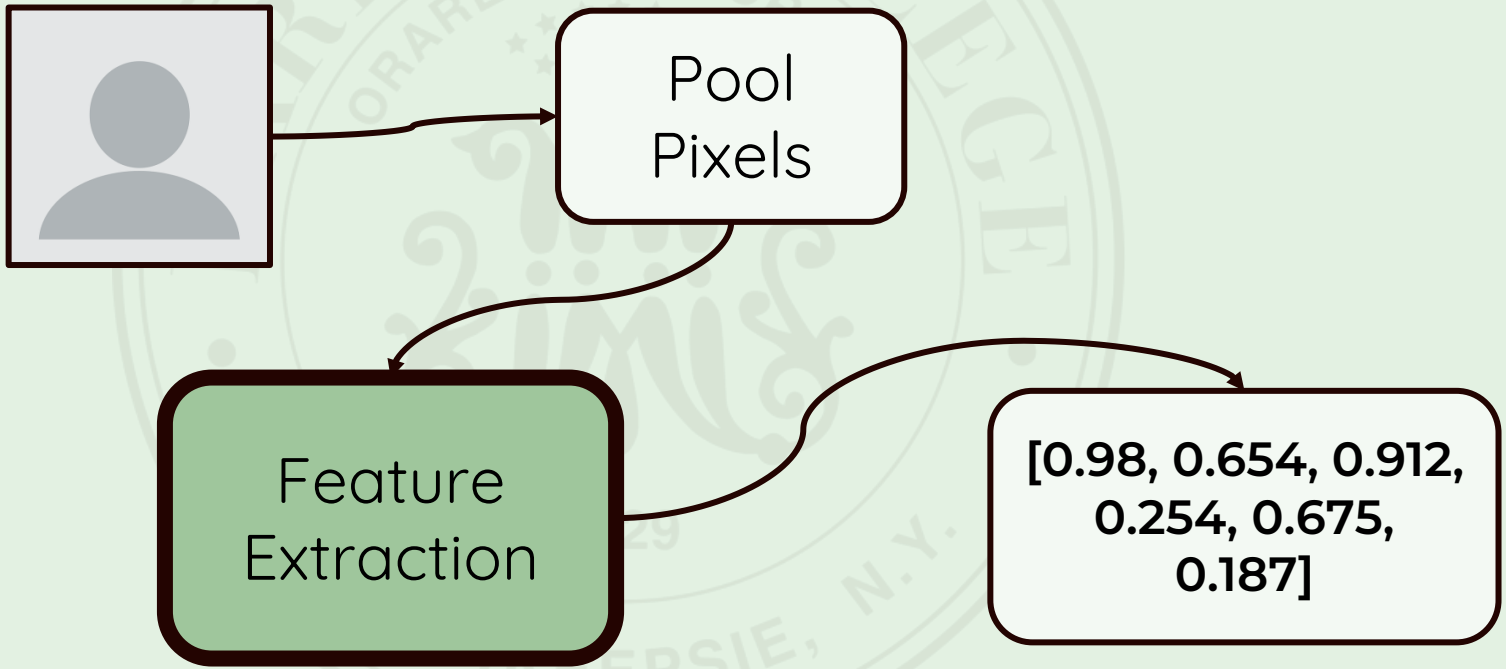
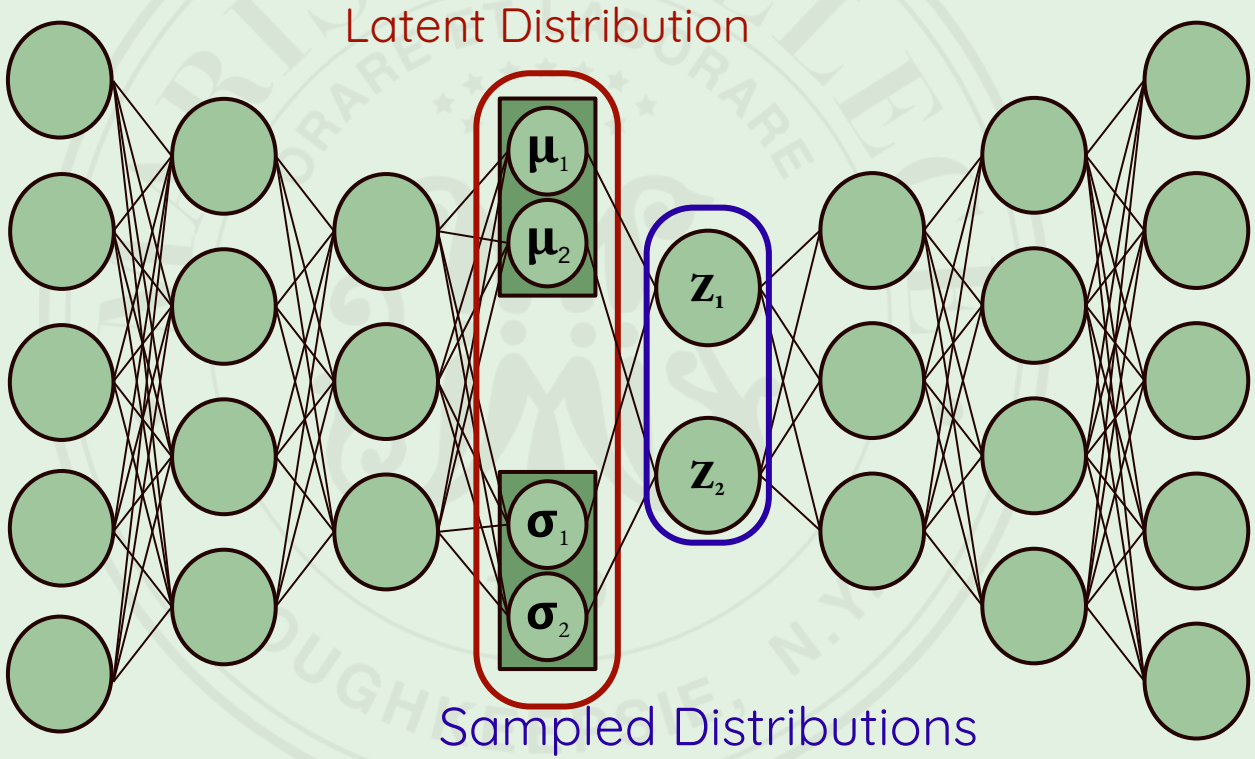


Figure 9

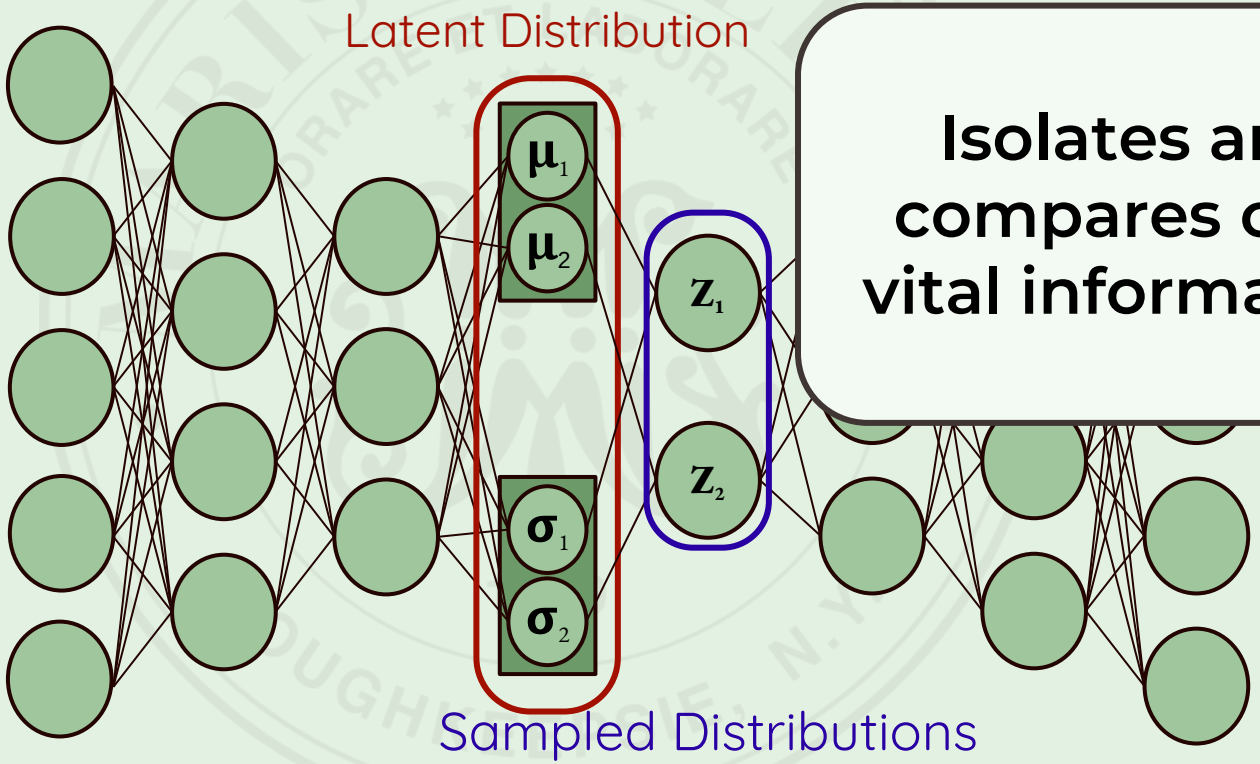
- **Variational Autoencoder**



(Rey et. al., 2021)

Figure 8

- **Variational Autoencoder**

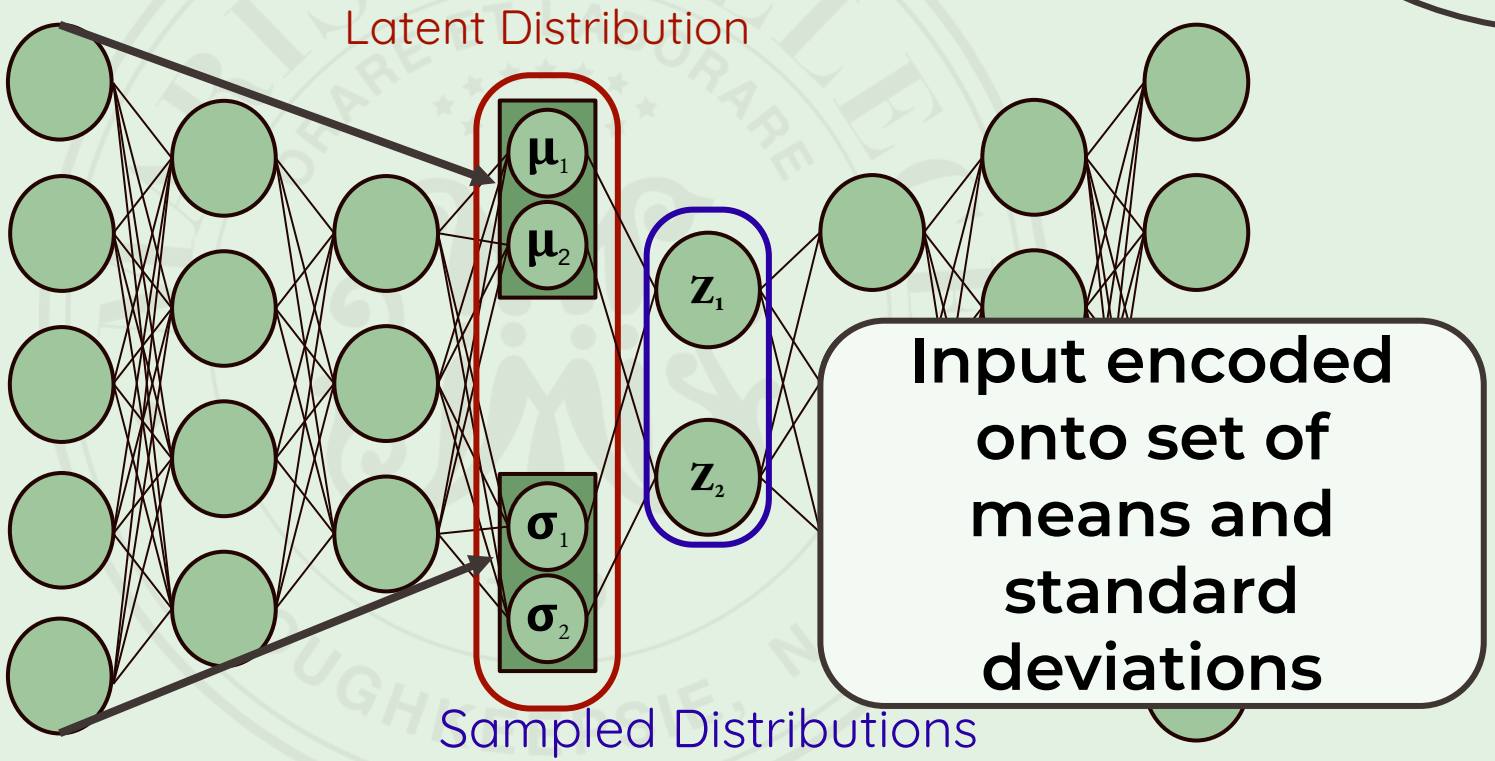


Isolates and compares only vital information

(Rey et. al., 2021)

Figure 8

- **Variational Autoencoder**

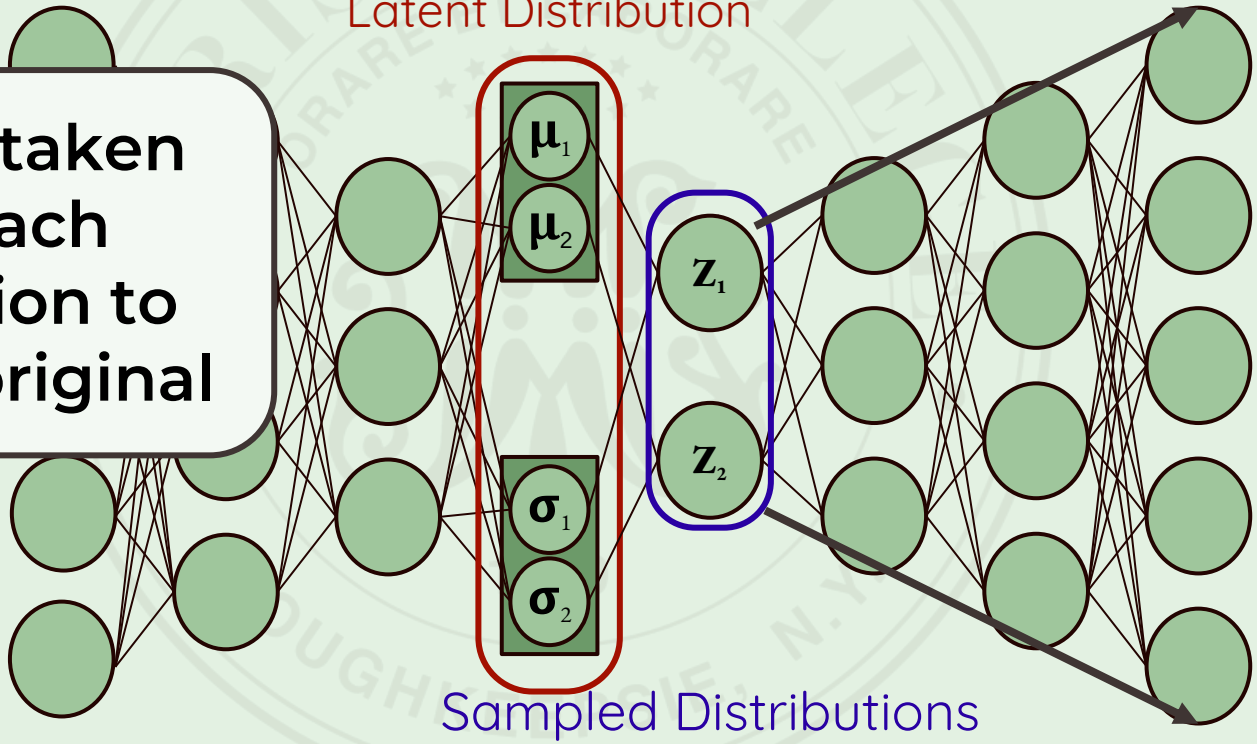


(Rey et. al., 2021)

Figure 8

- **Variational Autoencoder**

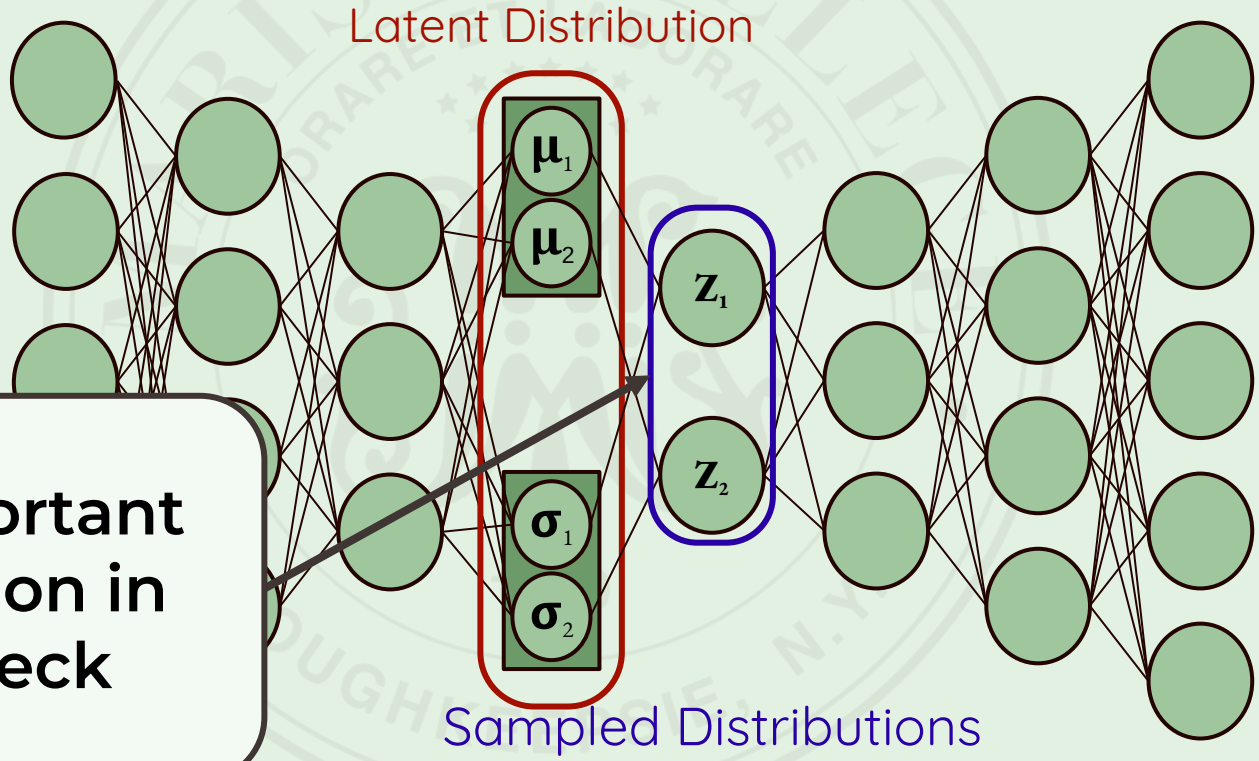
Samples taken from each distribution to recreate original



(Rey et. al., 2021)

Figure 8

- **Variational Autoencoder**

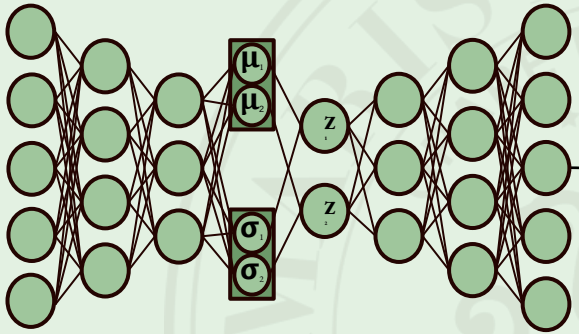


Only important information in bottleneck

Figure 8

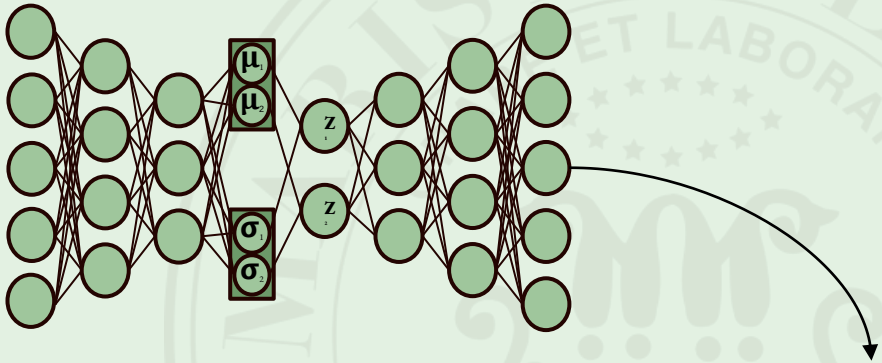
(Rey et. al., 2021)

- **Variational Autoencoder**



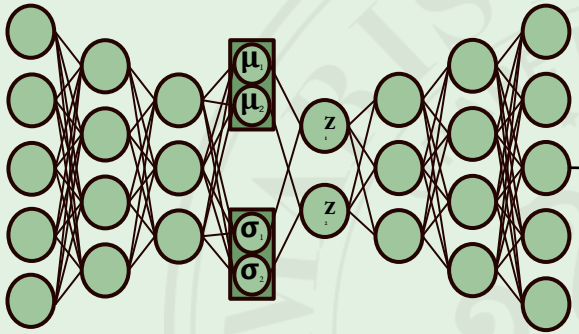
Mathematical interpretation

- **Variational Autoencoder**



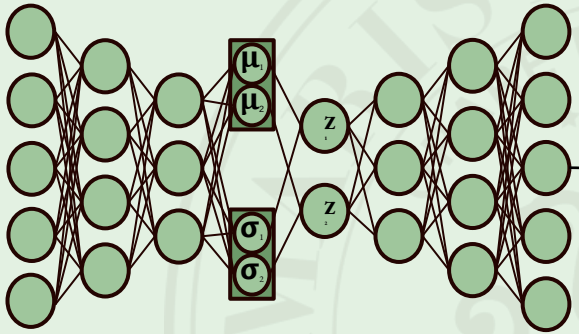
Given $x^{(i)}$ is a training input and W_i is the tensor of matrix multiplication weights

- # Variational Autoencoder



Given $x^{(i)}$ is a training input and W_i is the tensor of matrix multiplication weights
a(x) is an activation function: σ, tanh, ReLU etc.

- **Variational Autoencoder**



Given $x^{(i)}$ is a training input and W_i is the tensor of matrix multiplication weights

$a(x)$ is an activation function: σ, tanh, ReLU etc.

Basic auto encoder $f: R^{i,j} \rightarrow R^{i,j}$

$$f(x^{(i)}) = a(\dots a(a(x^{(i)} \cdot W_0) \cdot W_1) \cdot \dots \cdot W_{i-1}) =$$

- **Gradient-Based Optimization**

Given $f(x^{(i)}) = a(\ldots a(a(x^{(i)} \cdot W_0) \cdot W_1) \cdot \ldots \cdot W_{i-1}) = \hat{y}$

Gradient-Based Optimization

- Given $f(x^{(i)}) = a(\ldots a(a(x^{(i)} \cdot W_0) \cdot W_1) \cdot \ldots \cdot W_{i-1}) = \hat{y}$
- $C[f]$ is the cost $f(x^{(i)}) = \hat{y}$ between $\hat{y}$ and $x^{(i)}$

- **Gradient-Based Optimization**

Given $f(x^{(i)}) = a(\ldots a(a(x^{(i)} \cdot W_0) \cdot W_1) \cdot \ldots \cdot W_{i-1}) = \hat{y}$

$C[f]$ is the cost $f(x^{(i)}) = \hat{y}$

$f(x^{(i)}) = \hat{y}$ redefined as $f_w(W_i) = C[f]$

- **Gradient-Based Optimization**

  - **Given** $f_w(W_i) = C[f]$

Gradient-Based Optimization

- Given $f_w(W_i) = C[f]$
- $\nabla f_w(W_i)$ determines how to change W_i to maximize $C[f]$

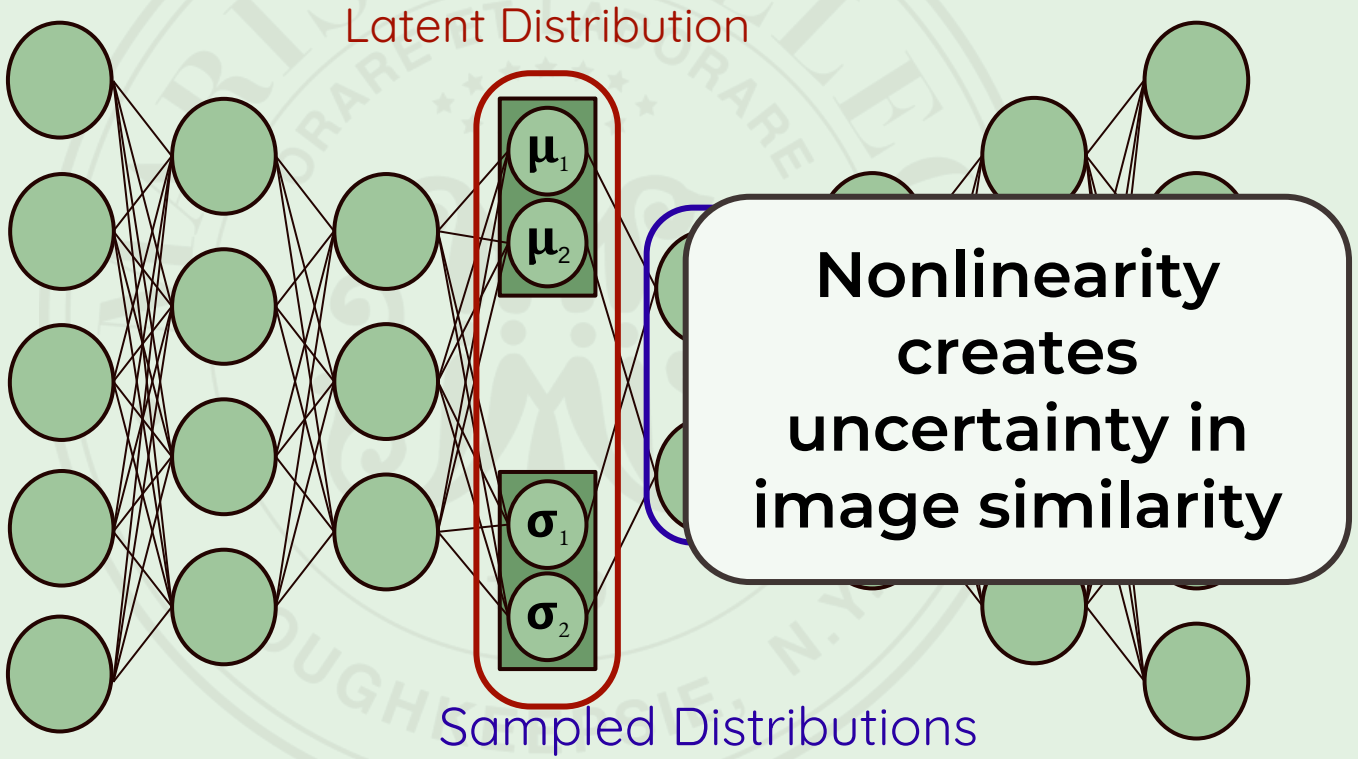
Gradient-Based Optimization

Given $f_w(W_i) = C[f]$

$-\nabla f_w(W_i)$ determines how to change W_i to minimize $C[f]$

$W_i - \nabla f_w(W_i)$ optimizes $f(x^{(i)})$

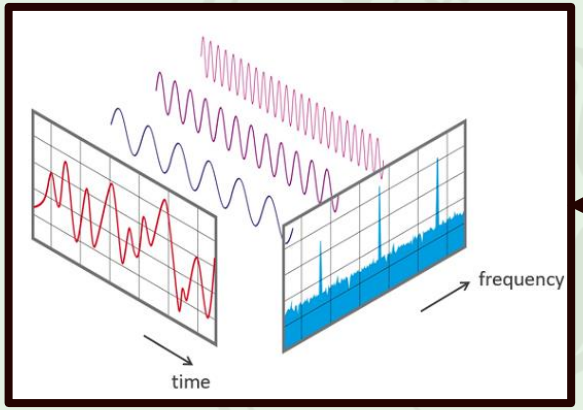
- **Variational Autoencoder**



(Rey et. al., 2021)

Figure 8

- **Perceptual Hashing** (Koul, et. al., 2009)



Feature Extraction

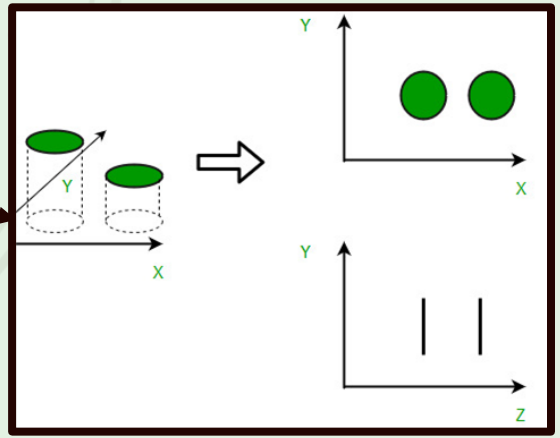
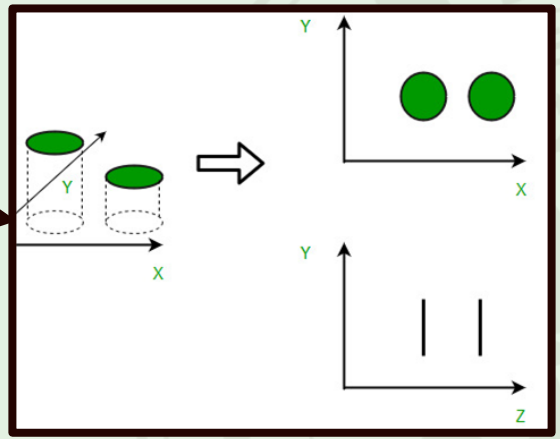


Figure 9

0.137]

- **Dimensionality Reduction**



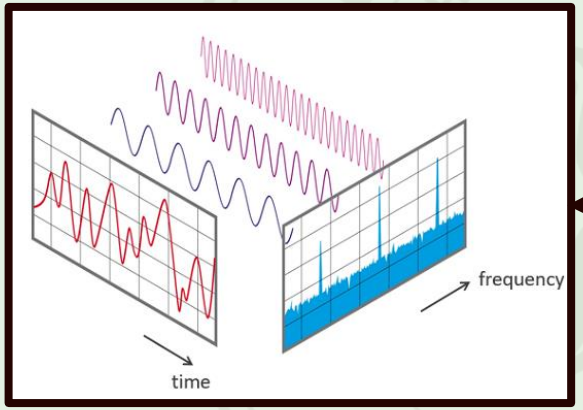
Reduces the amount for information per datapoint

Takes advantage of the **Manifold Theory**

All data lies on a smooth and continuous lower dimensional manifold in the higher dimensional space of all possible

Assumption allow for most reduction

- **Perceptual Hashing** (Koul, et. al., 2009)



Feature Extraction

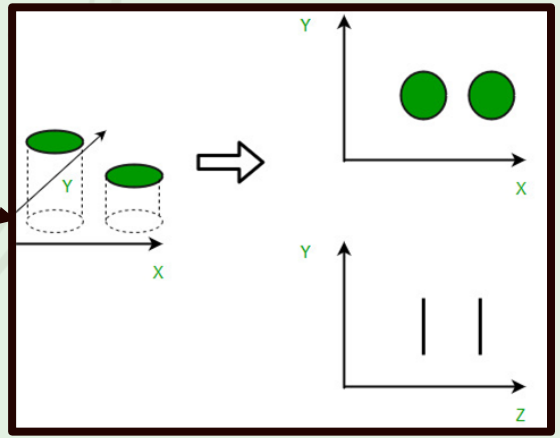
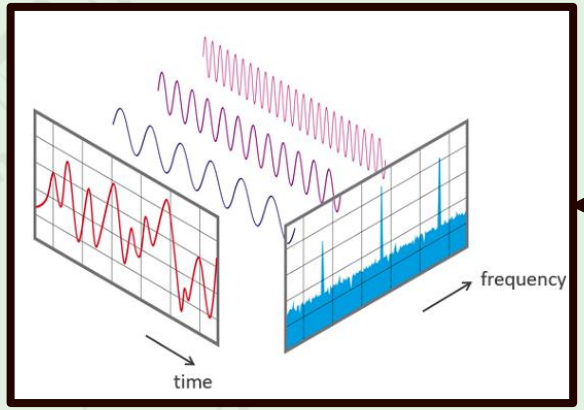


Figure 9

Frequency Domain Transform

Converts from wave to a frequency domain

Fourier Transform, Discrete Cosine Transform **(DCT)**



(Koul, et. al., 2009)

Focuses on pattern of information so if image is scaled or tone is changed, will not affect DCT

- **Perceptual Hashing** (Koul, et. al., 2009)

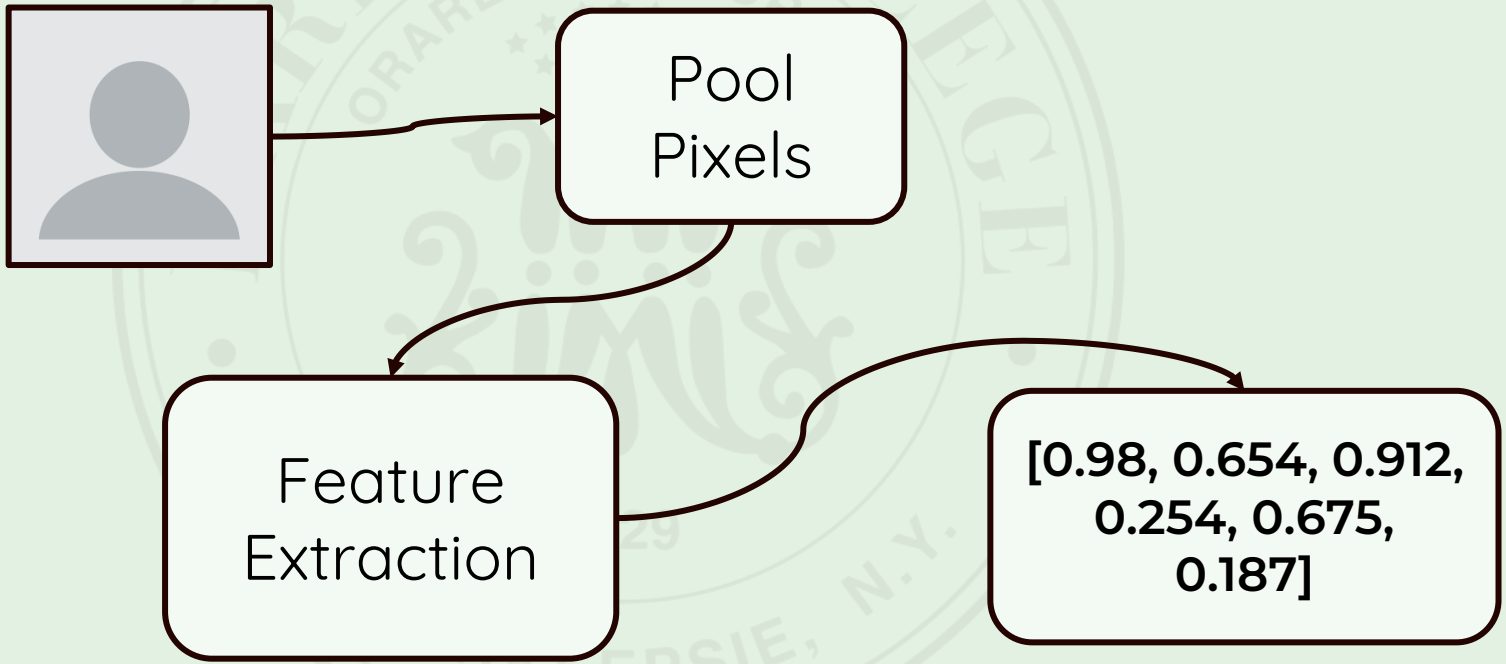


Figure 9

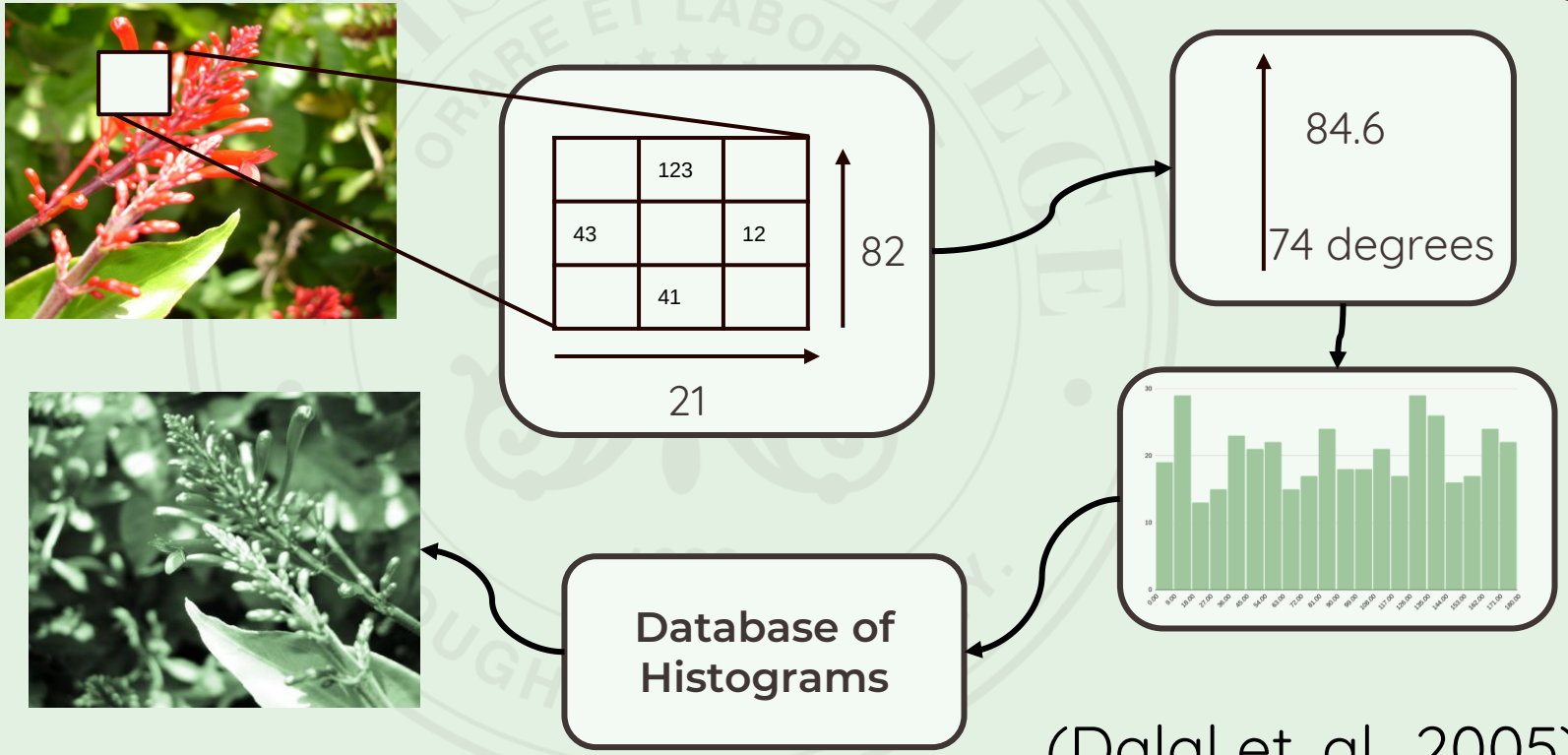
- **Texton Algorithms**

- **Texton Algorithms**

- **Texton Algorithms**

Analyzes color/textures in individual textons

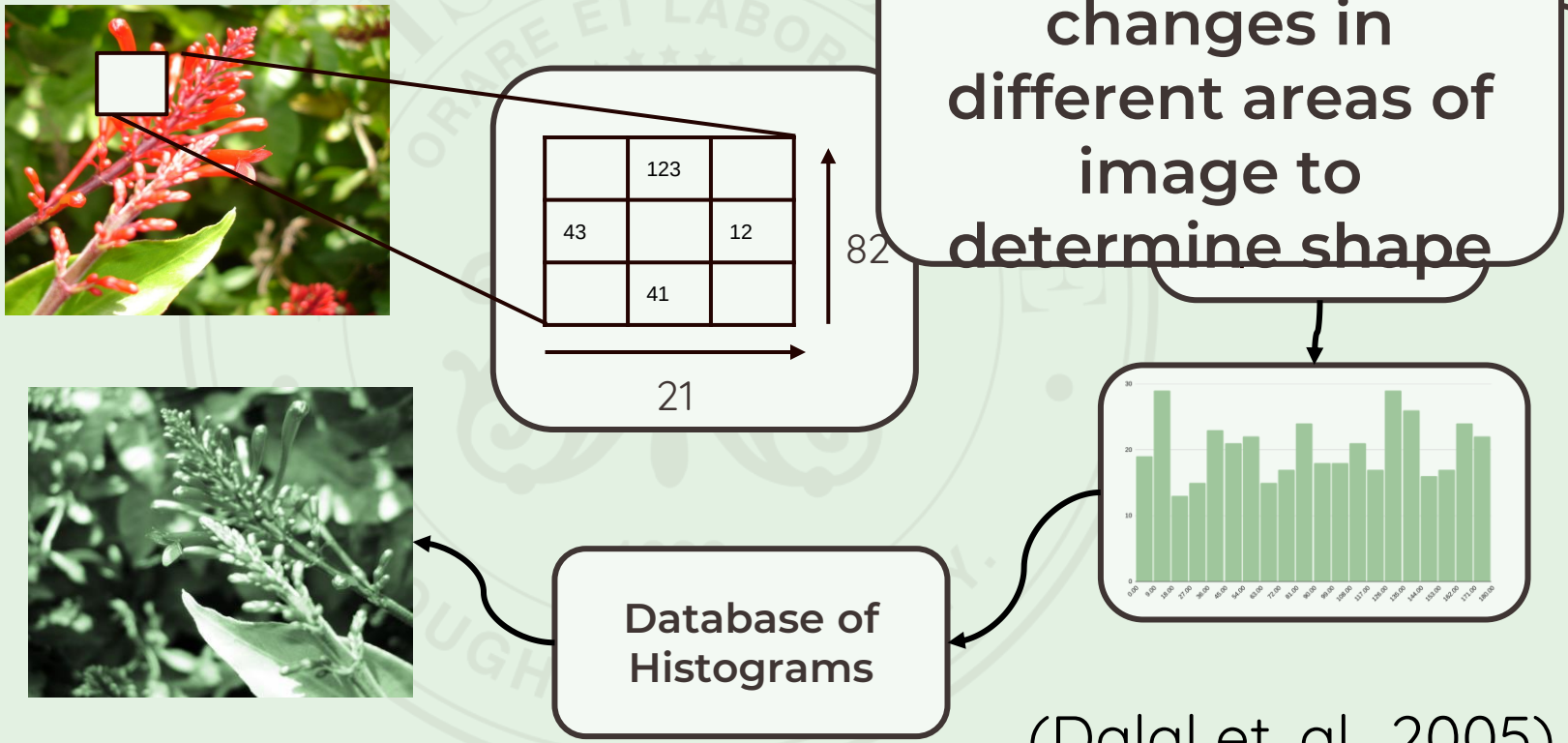
Histogram of Oriented Gradients



(Dalal et. al., 2005)

Figure 10

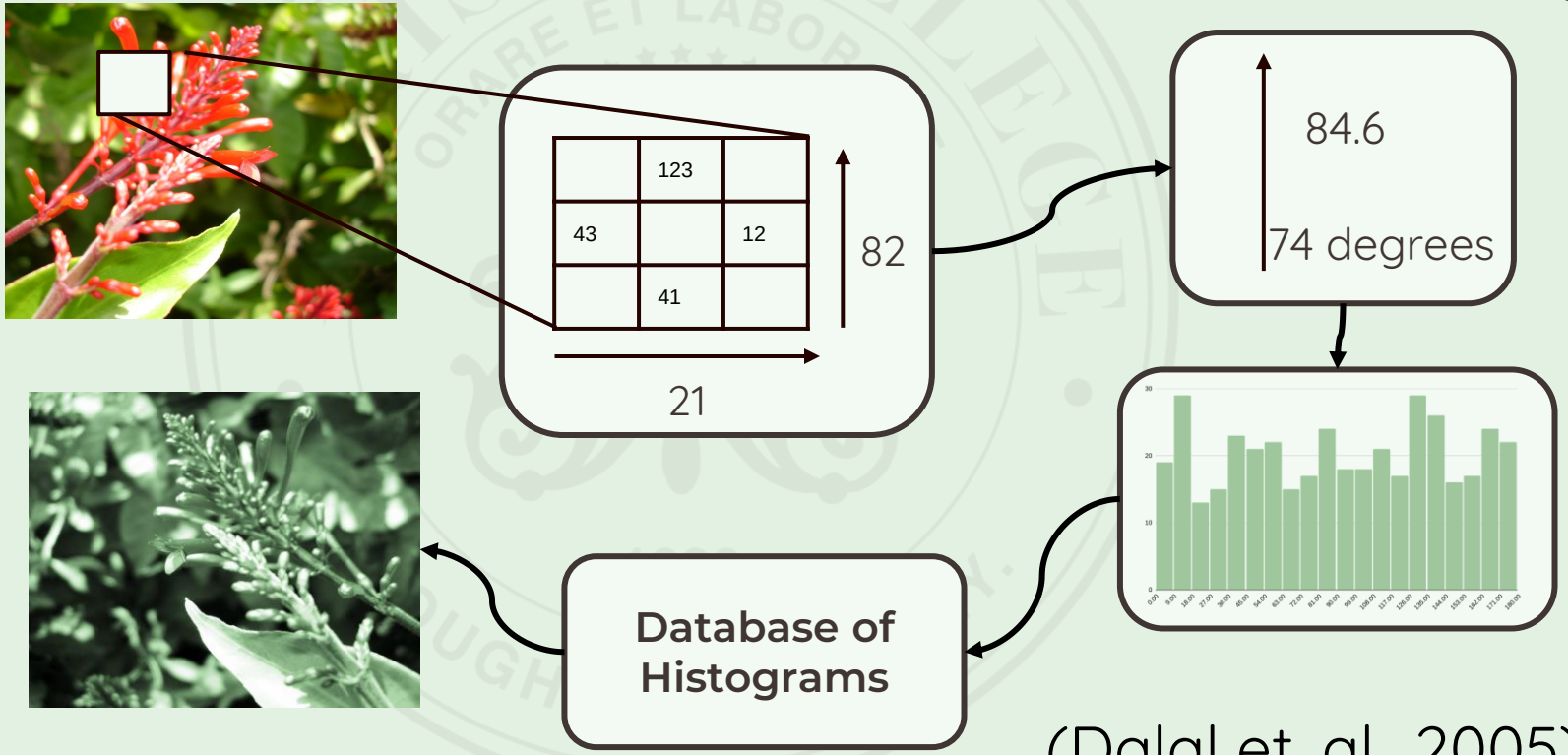
- **Histogram of Oriented Gradients**



(Dalal et. al., 2005)

Figure 10

Histogram of Oriented Gradients



(Dalal et. al., 2005)

Figure 10

Keypoint Algorithms

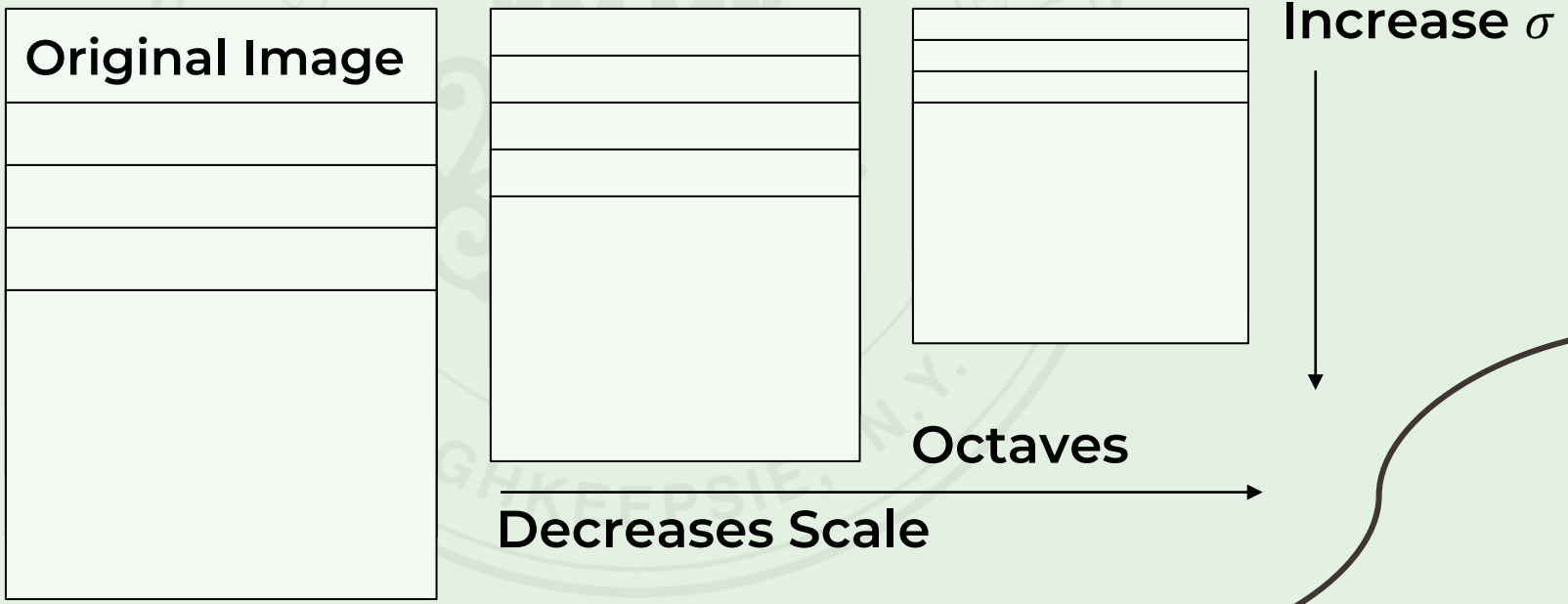
Generate a tensor of keypoints and descriptors that describe image

Keypoints and descriptors are then referenced from database to find similar clusters

Key points must be invariant to different perspectives

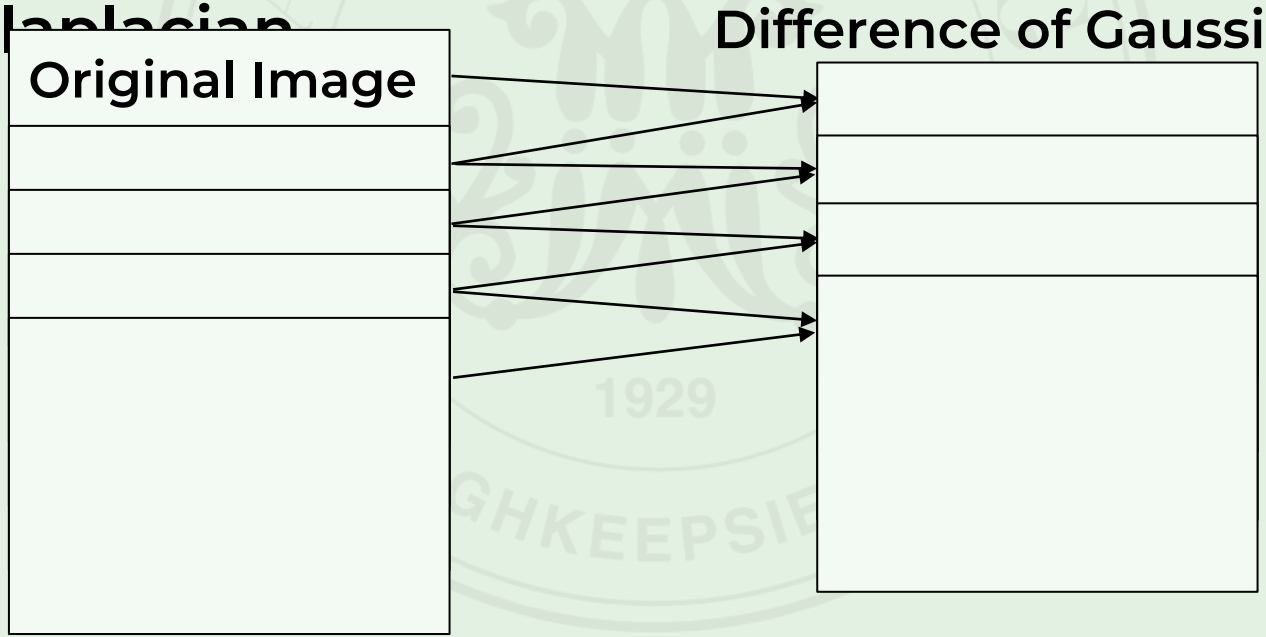
Scale Invariant Feature Transform

In order to ensure scale invariance first transforms image to a scale space L(x,y,σ)



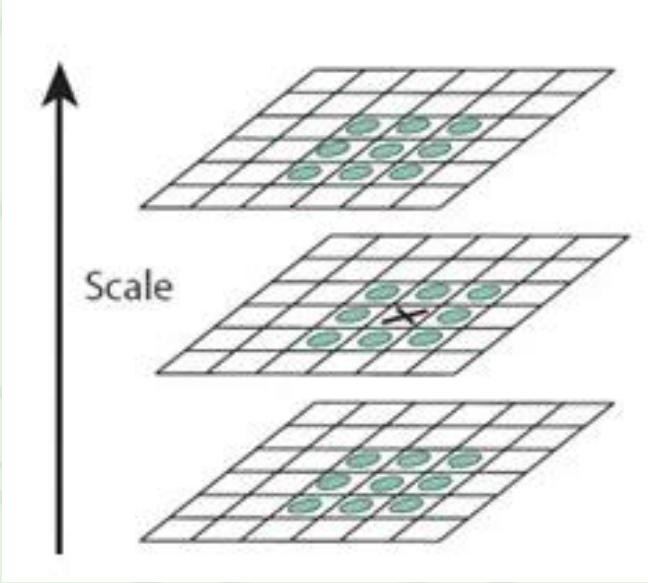
Scale Invariant Feature Transform

Difference of different layers within an octave used to approximate difference of laplacian



- **Scale Invariant Feature Transform**

Key points are defined as the local extrema between scales



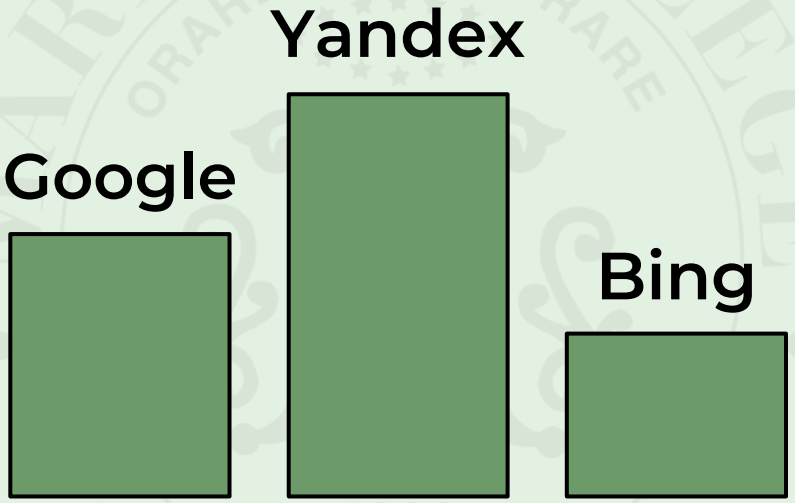
Scale Invariant Feature Transform

Extrema, which lie on an edge or do not have sufficient contrast, are removed

Extrema locations are refined through Taylor expansion of scale space

HOG around key points used to ensure rotational invariance

- **Current Top Algorithms**



(Toler, 2018)

- **Issues with Toler Study**

- **Issues with Toler Study**

  - Very few images used

- **Issues with Toler Study**
  - Very few images used
  - **Only cluttered landscape images used**

Methods

Figure 5

Methods

105 test images collected

Figure 5

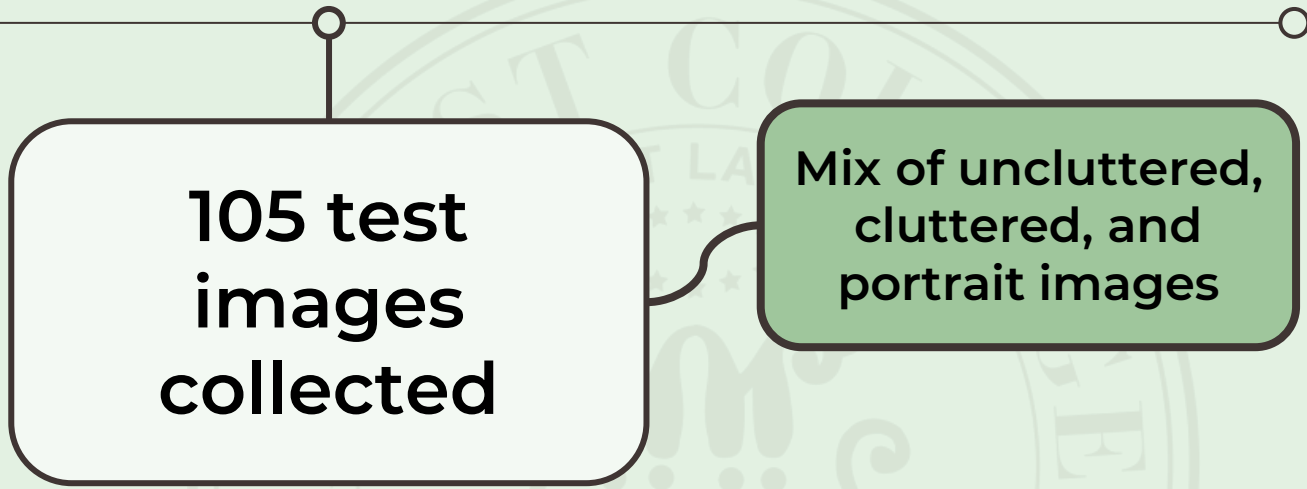

105 test images collected
Mix of uncluttered, cluttered, and portrait images

Figure 5

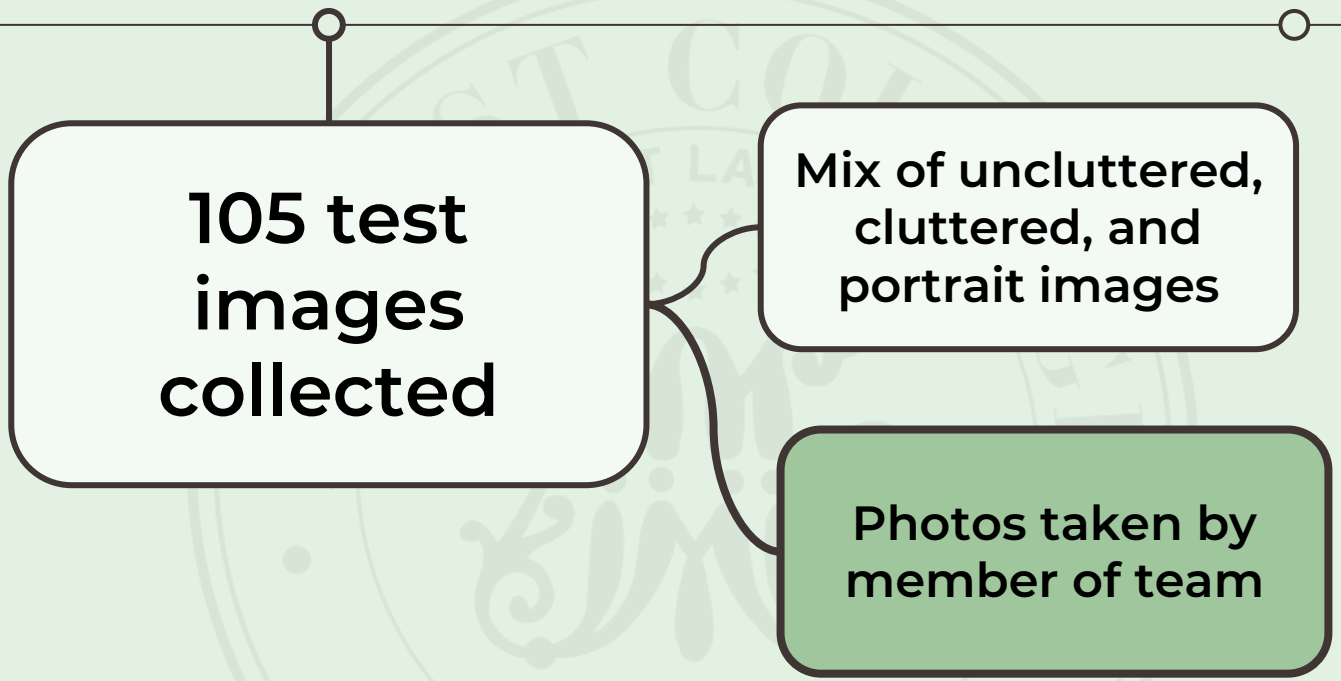

105 test images collected
Mix of uncluttered, cluttered, and portrait images
Photos taken by member of team

Figure 5

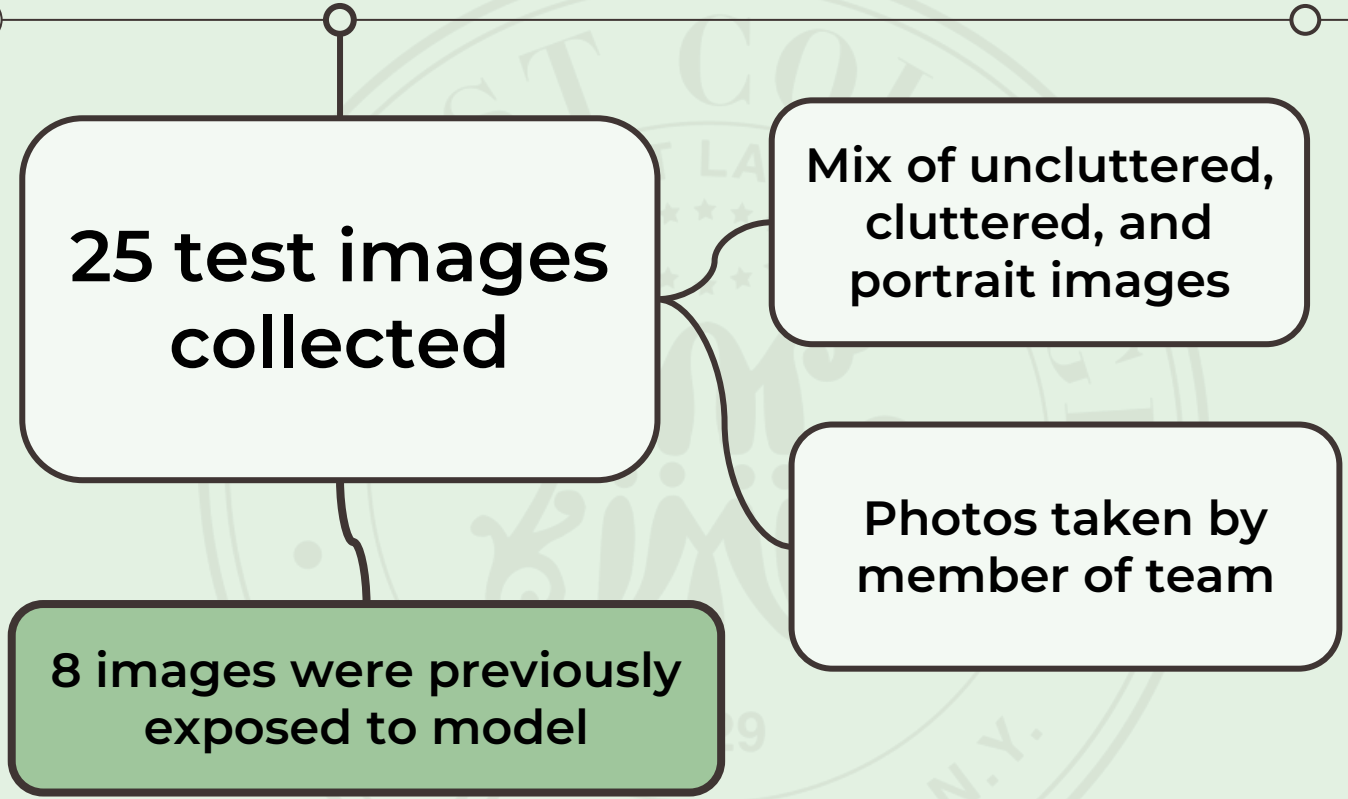
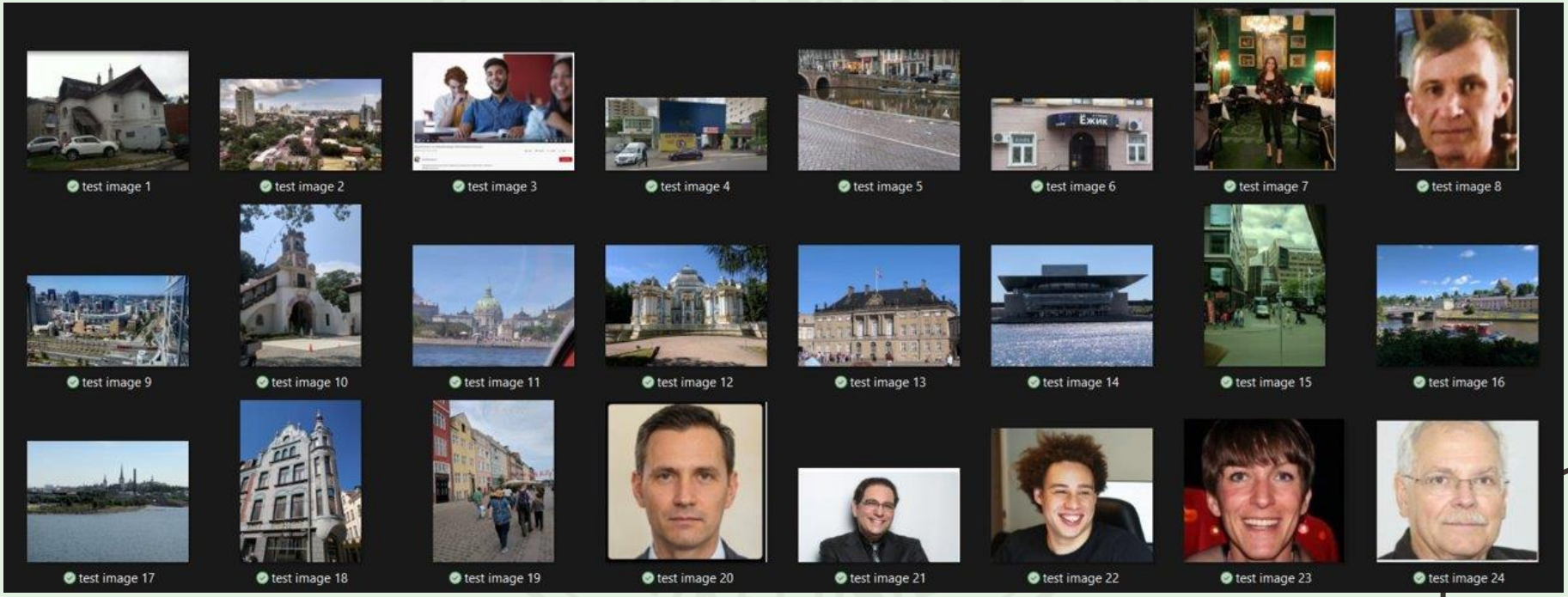

25 test images collected
Mix of uncluttered, cluttered, and portrait images
Photos taken by member of team
8 images were previously exposed to model

Figure 5

Test Images


test image 1
test image 2
test image 3
test image 4
test image 5
test image 6
test image 7
test image 8
test image 9
test image 10
test image 11
test image 12
test image 13
test image 14
test image 15
test image 16
test image 17
test image 18
test image 19
test image 20
test image 21
test image 22
test image 23
test image 24

Methods

25 test images collected

Input to reverse image search algorithm

Figure 5

test images
collected

Input to reverse image search algorithm

Google, Bing, Yandex chosen as algorithms

Figure 5

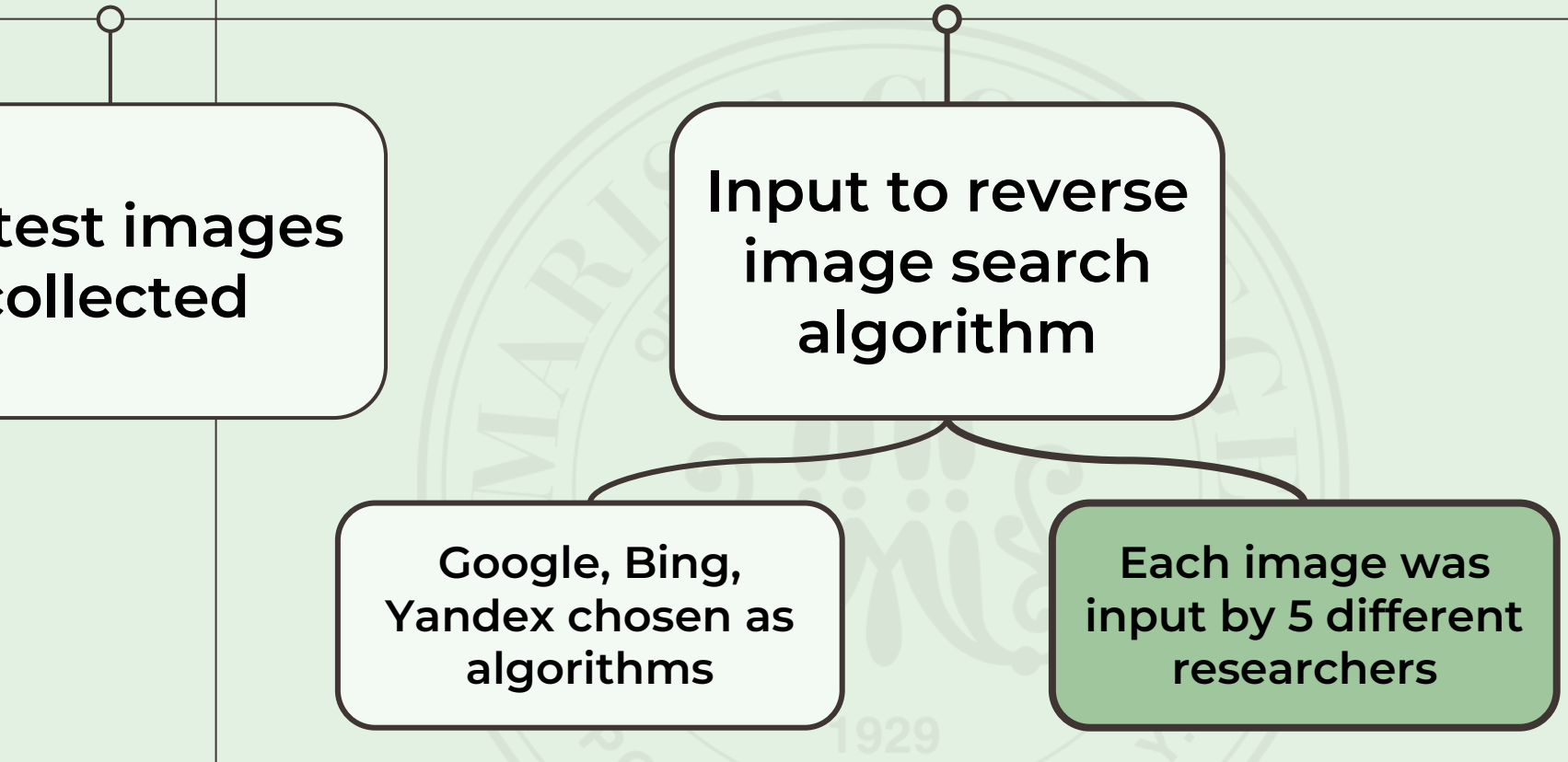

test images
collected
Input to reverse image search algorithm
Google, Bing, Yandex chosen as algorithms
Each image was input by 5 different researchers

Figure 5

Methods

105 test images collected

Input to reverse image search algorithm

Reverse Image search results analyzed

Figure 5

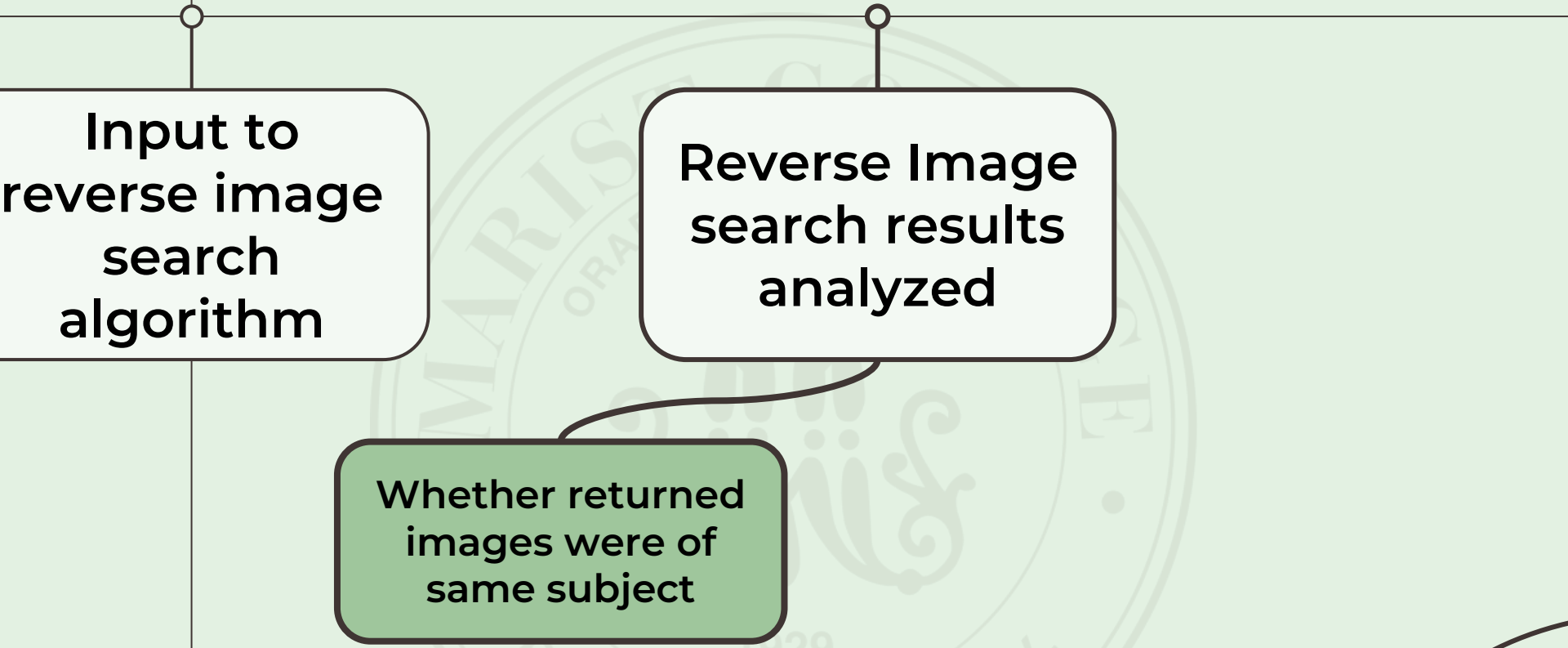

Input to reverse image search algorithm
Reverse Image search results analyzed
Whether returned images were of same subject

Figure 5

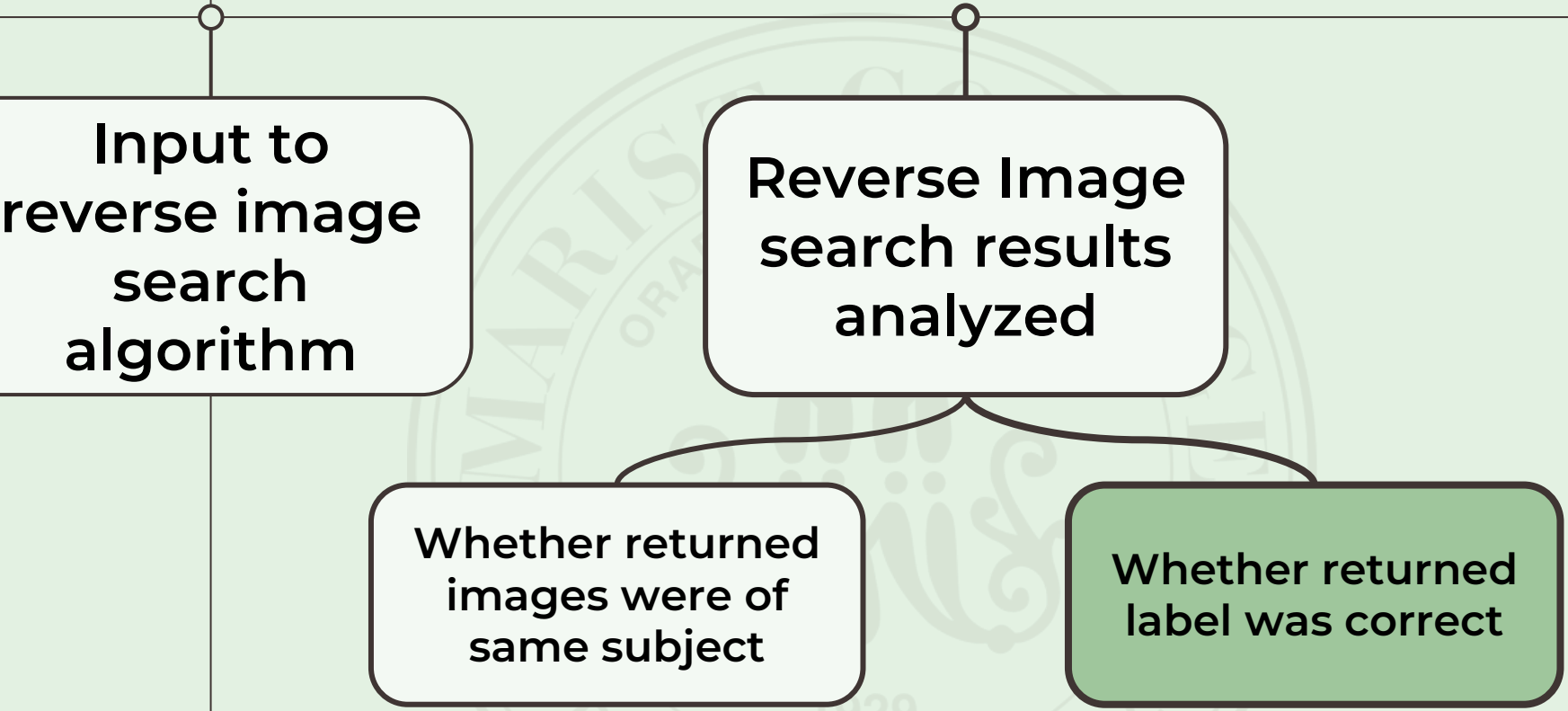

Input to reverse image search algorithm
Reverse Image search results analyzed
Whether returned images were of same subject
Whether returned label was correct

Figure 5

- **Performance of Algorithms**

Figure 13

Performance of Algorithms

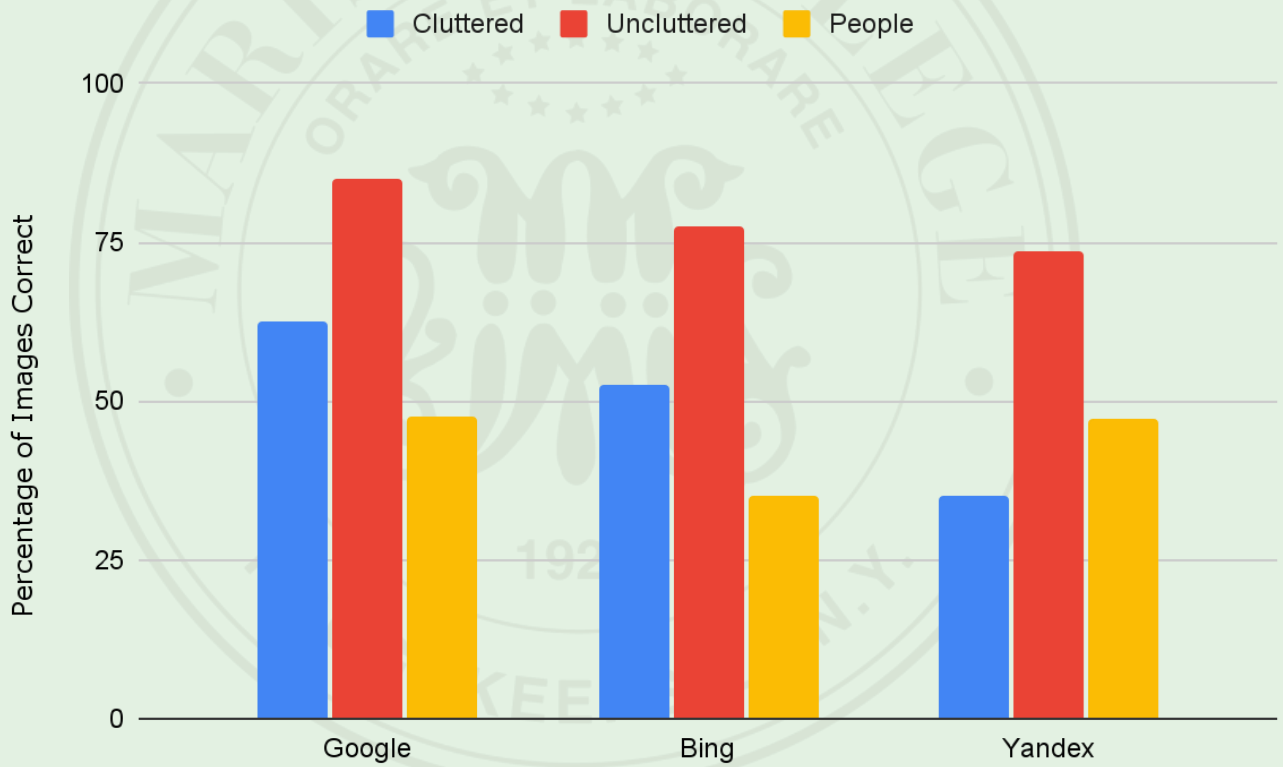
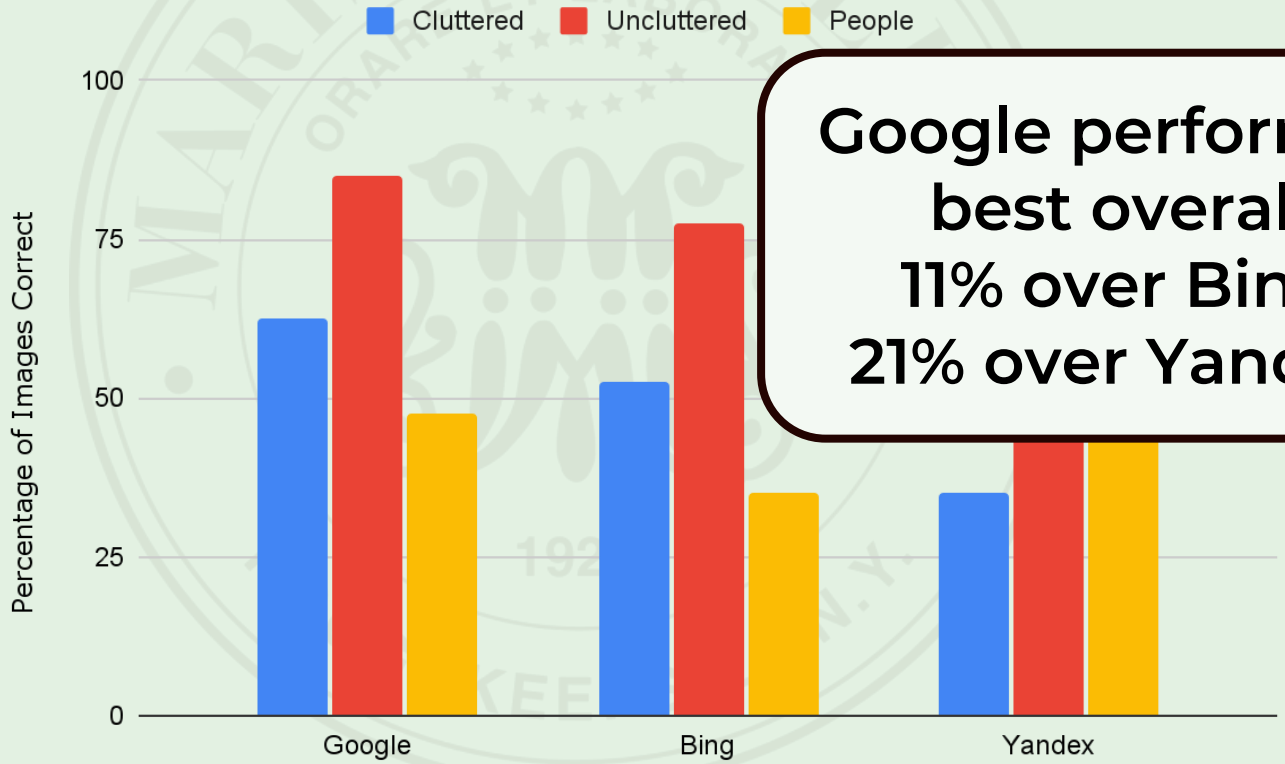


Figure 13

Performance of Algorithms



Google performed best overall
11% over Bing
21% over Yandex

Figure 13

- **Performance of Algorithms**

Yandex performed significantly worse on cluttered images

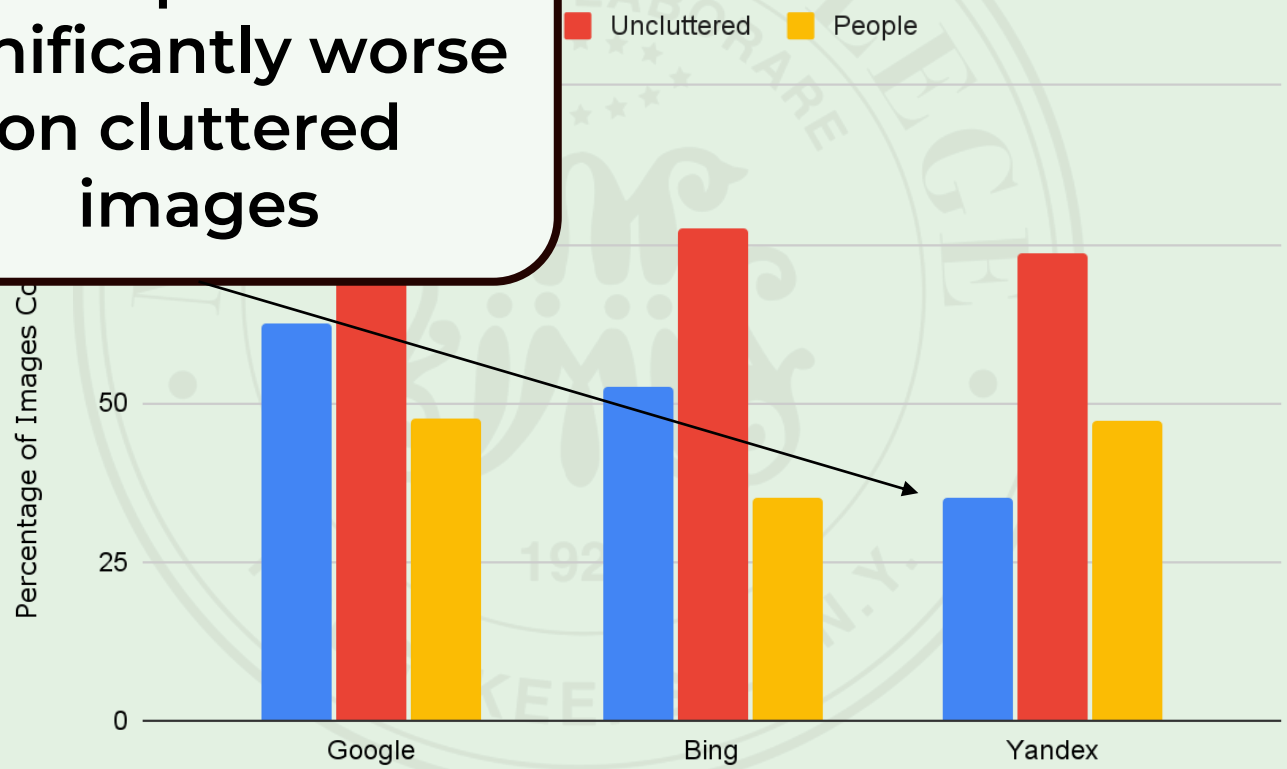


Figure 13

- **Performance of Algorithms**

Uses basic CBIR algorithms

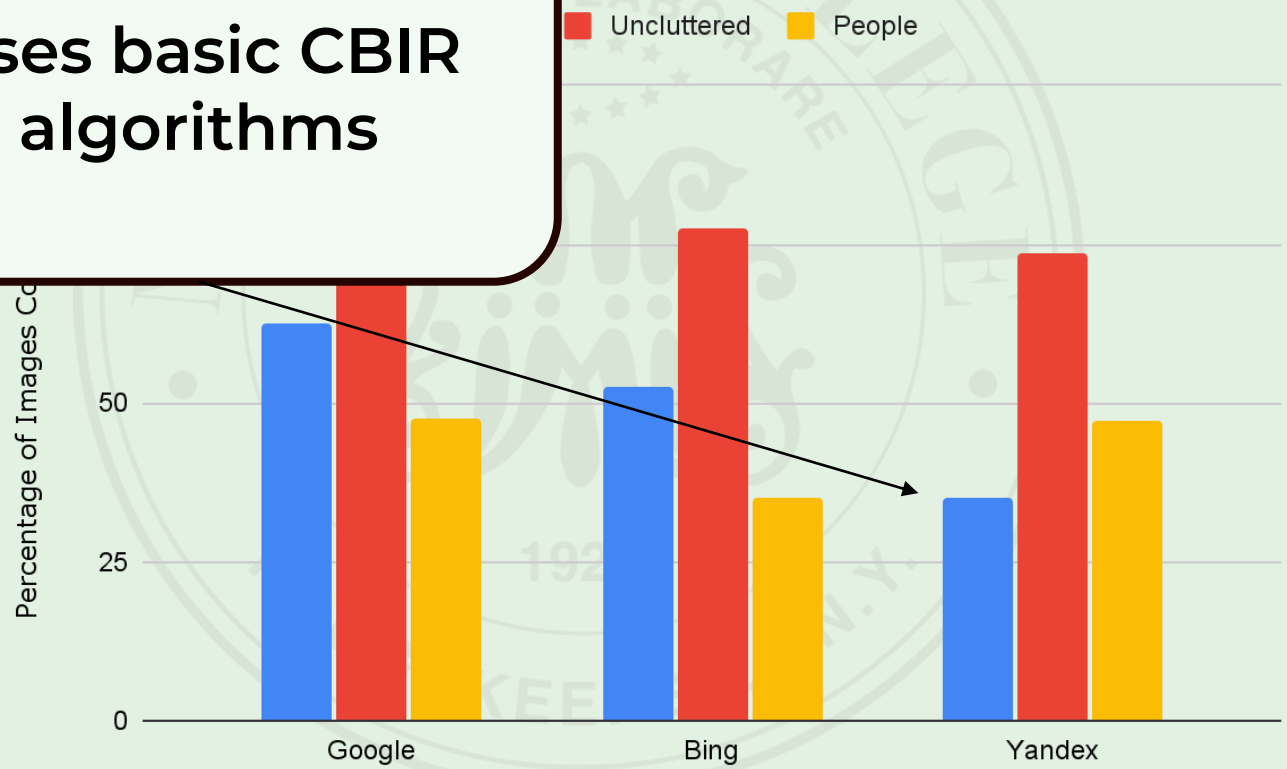


Figure 13

- **Performance of Algorithms**

Bing performed 11% worse on facial recognition

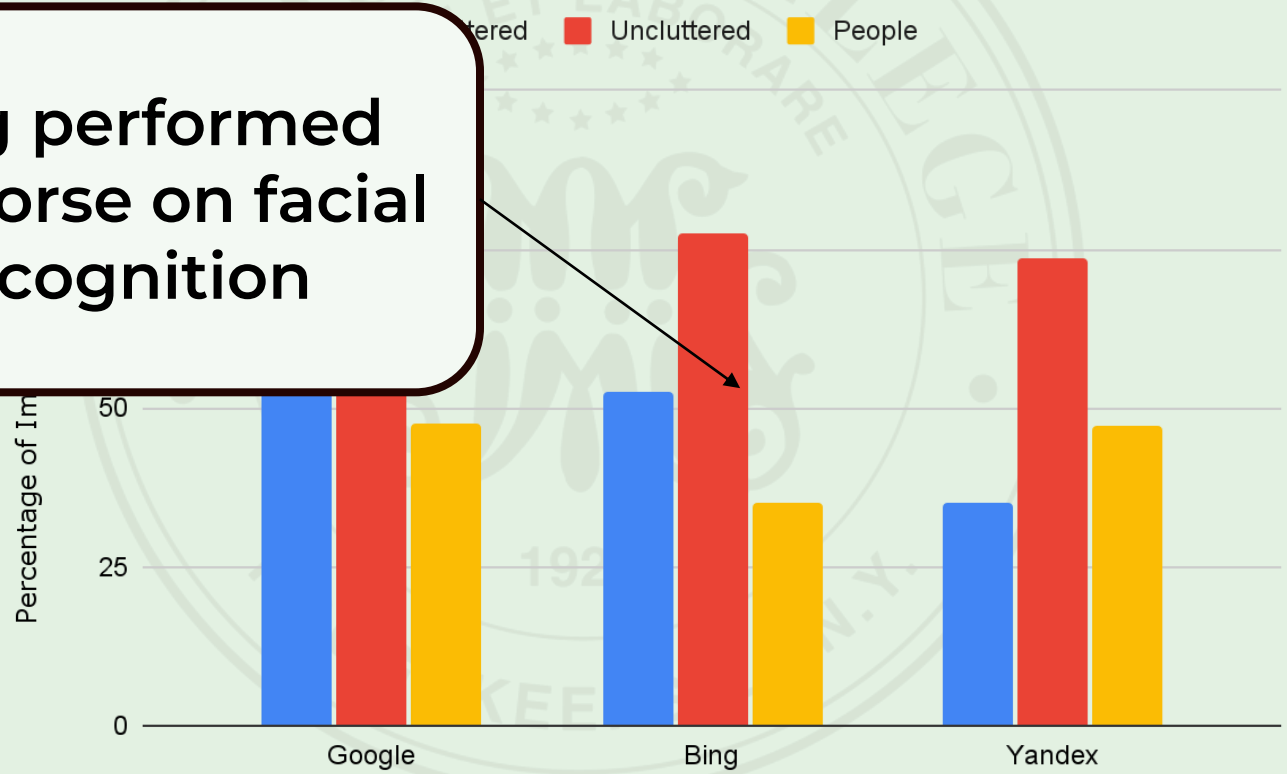


Figure 13

Performance of Algorithms

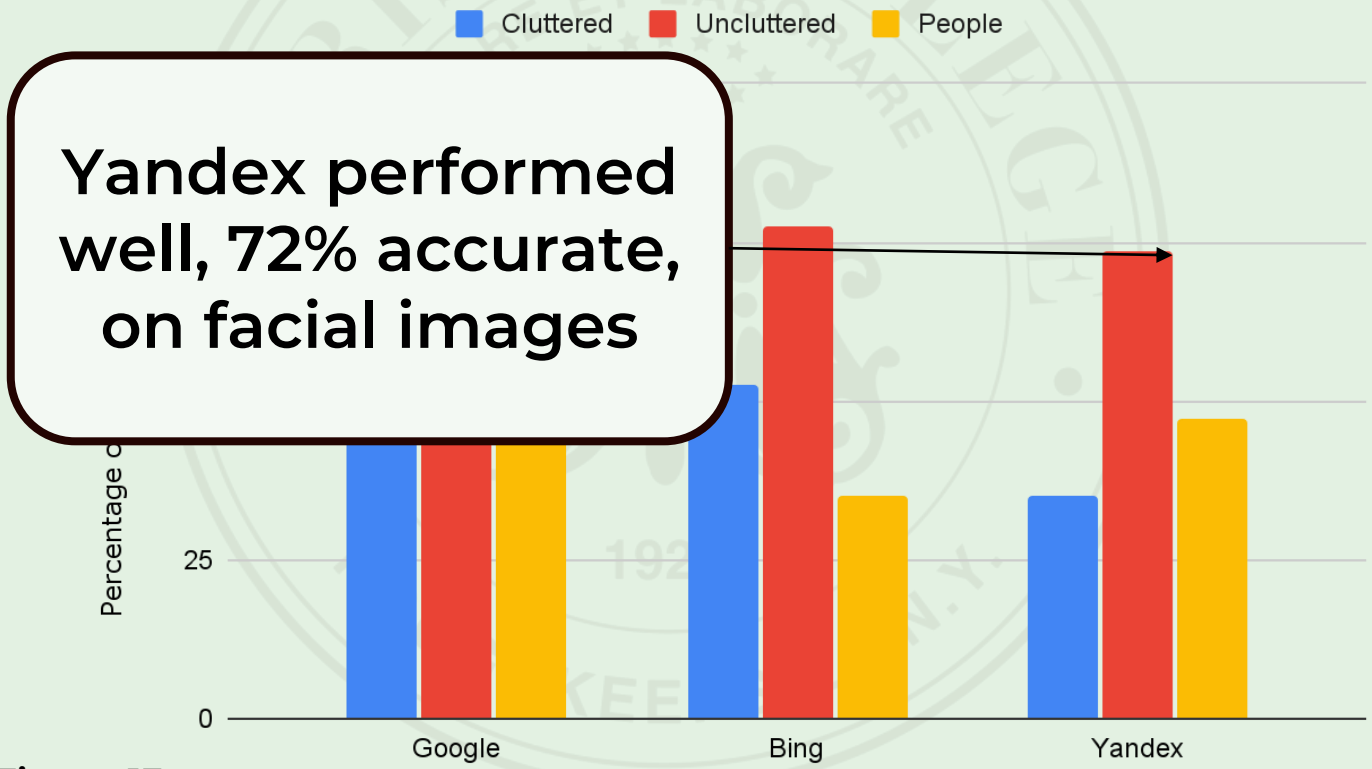


Figure 13

Performance On Exposed Images

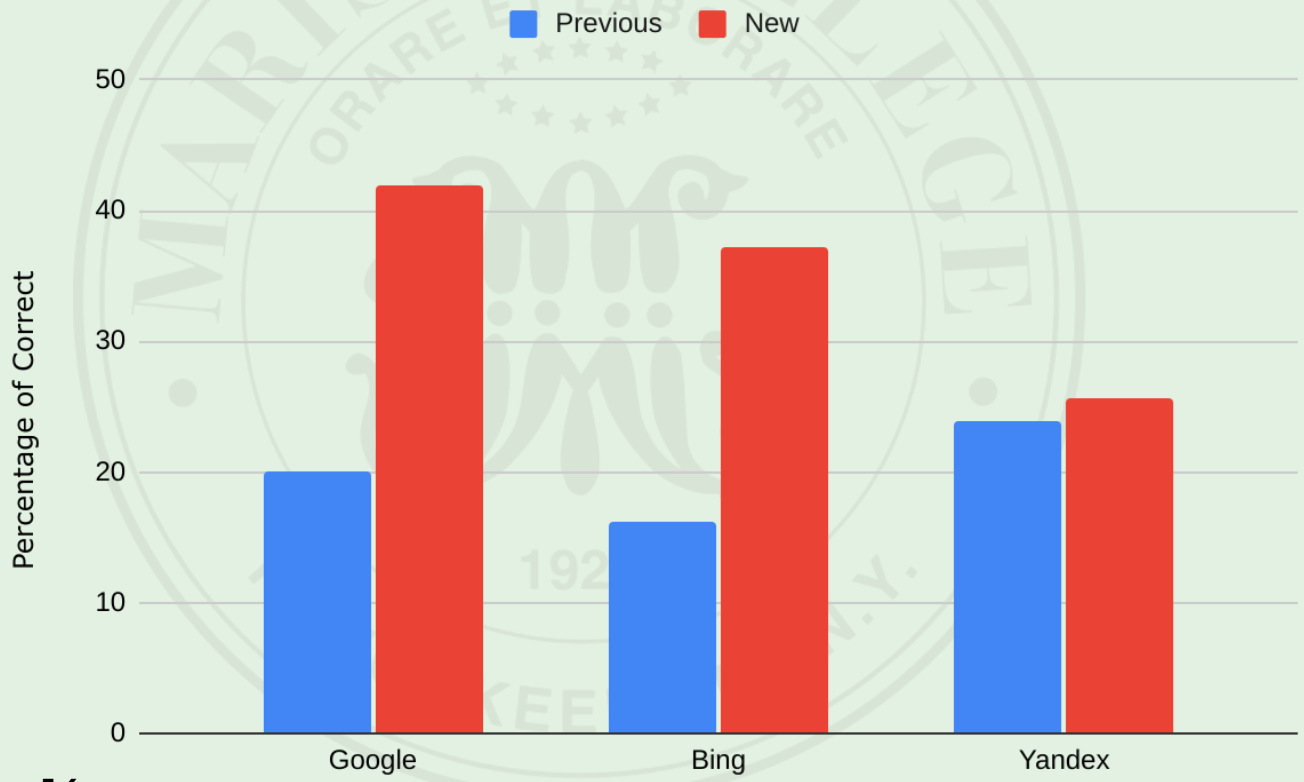


Figure 14

Performance On Exposed Images

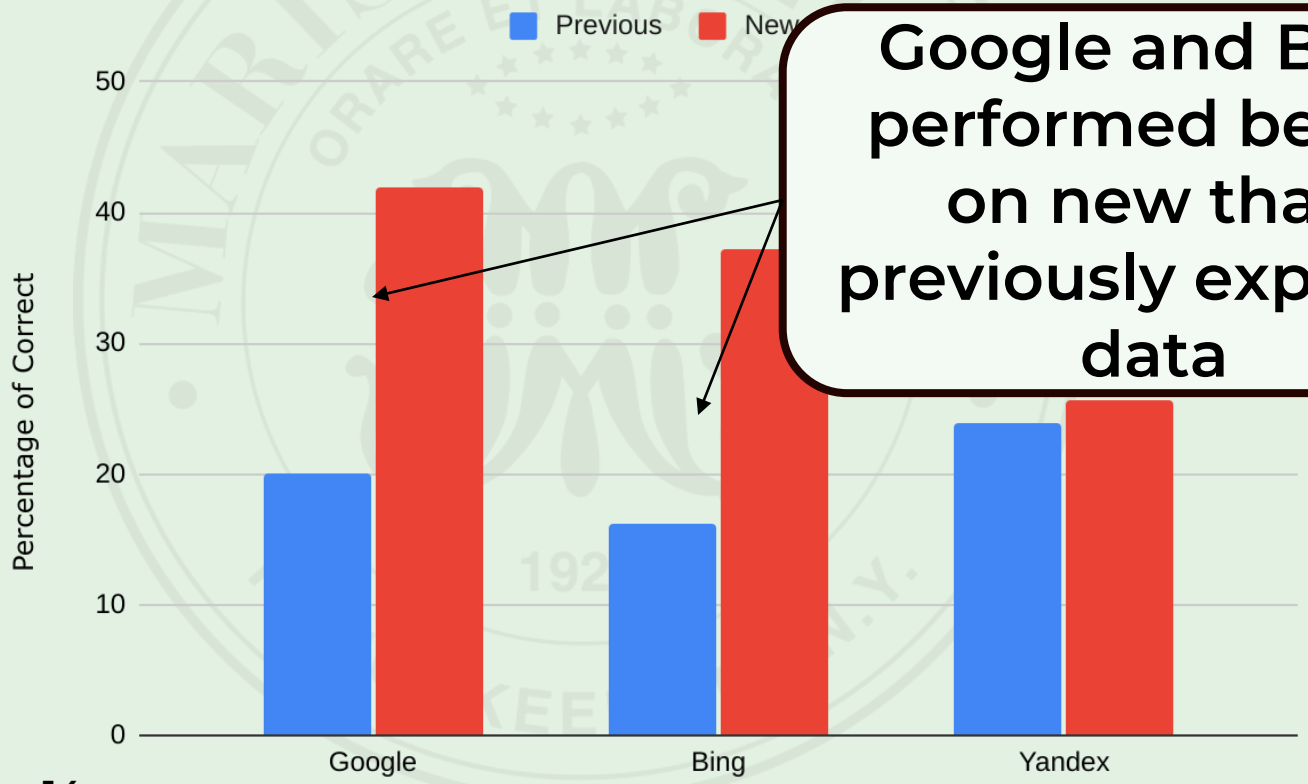


Figure 14

Performance On Exposed Images

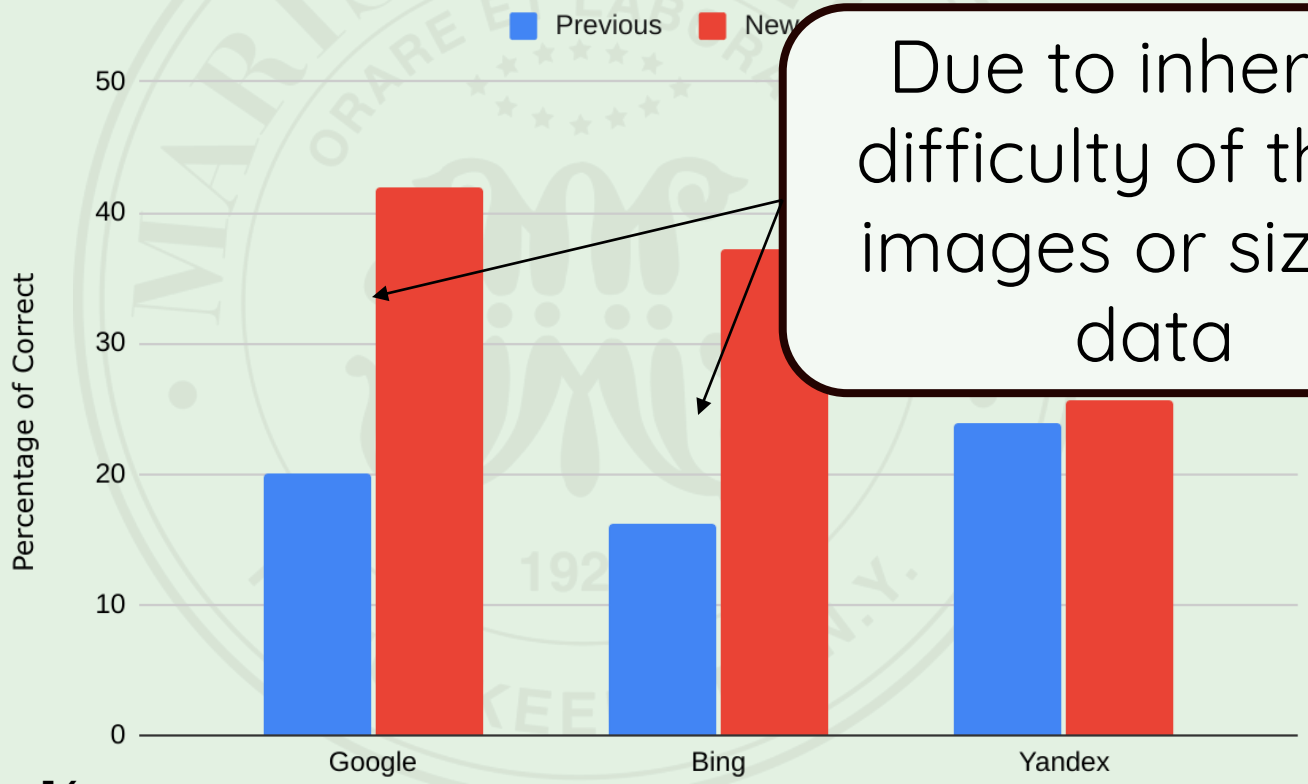


Figure 14

- # Performance Before/After Update

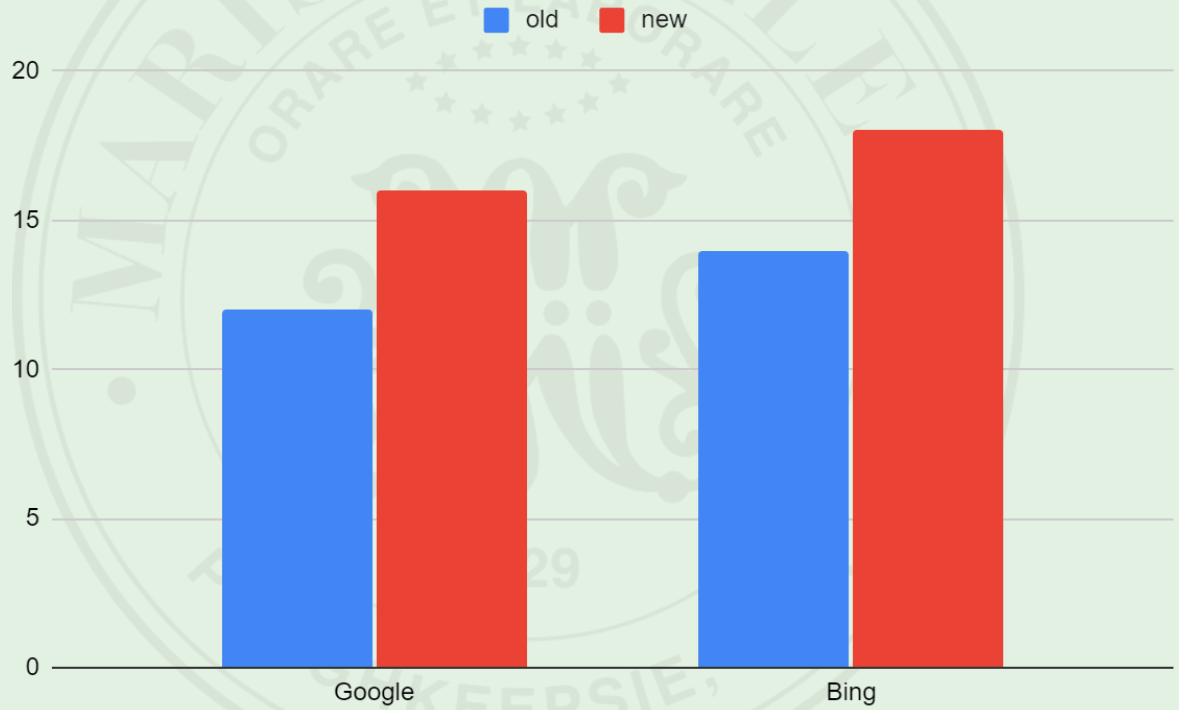


Figure 15

Google and Bing Both showed significant improvement

fore/After Update

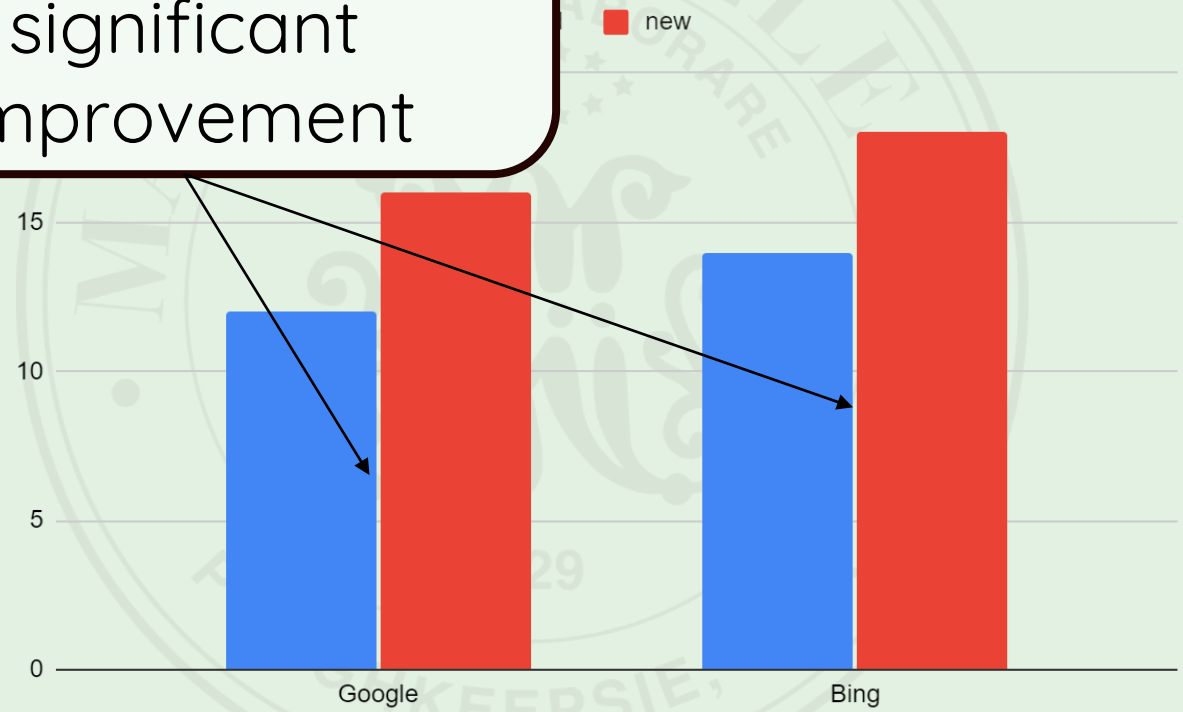
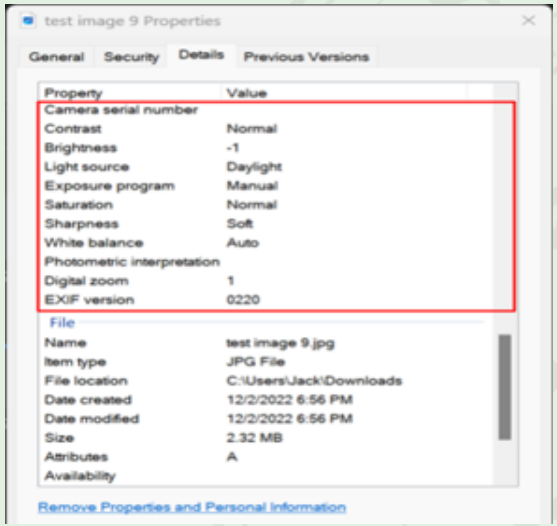


Figure 15

EXIF Data Analysis



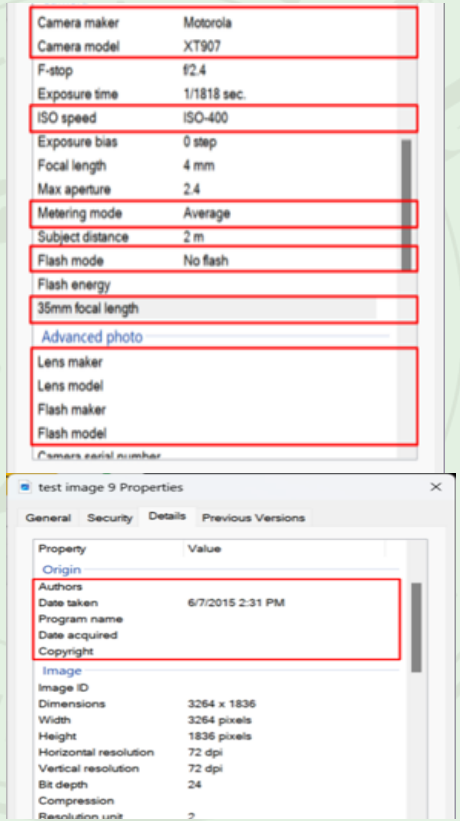
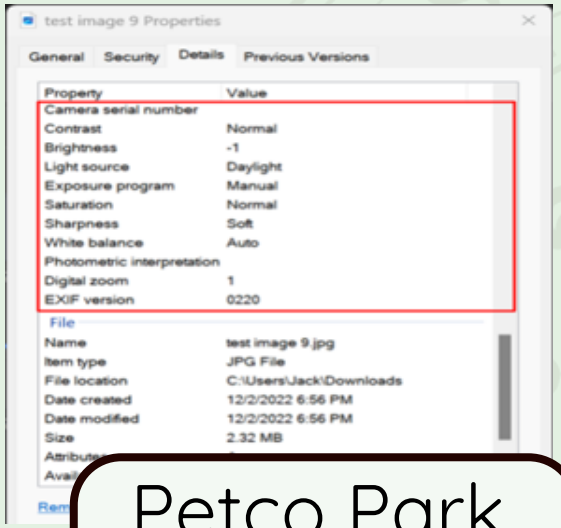
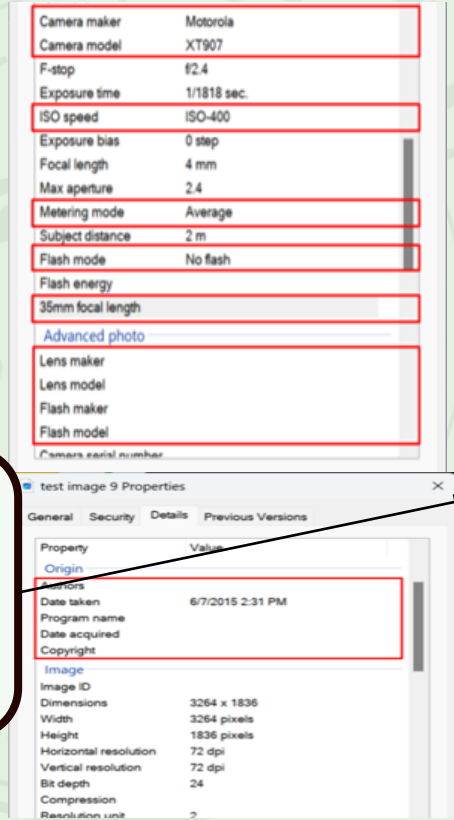


Image: Test Image 9

EXIF Data Analysis

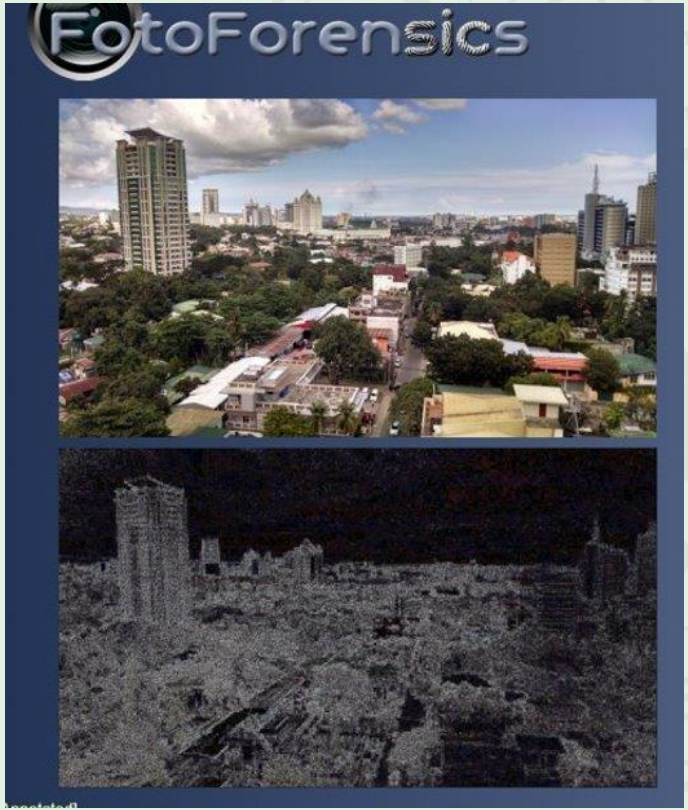


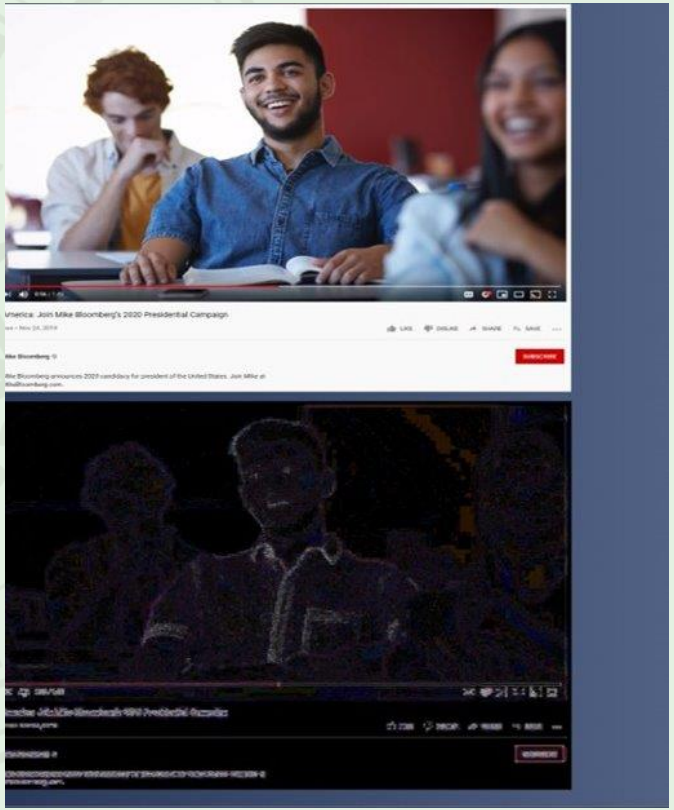


Petco Park, San Diego California.

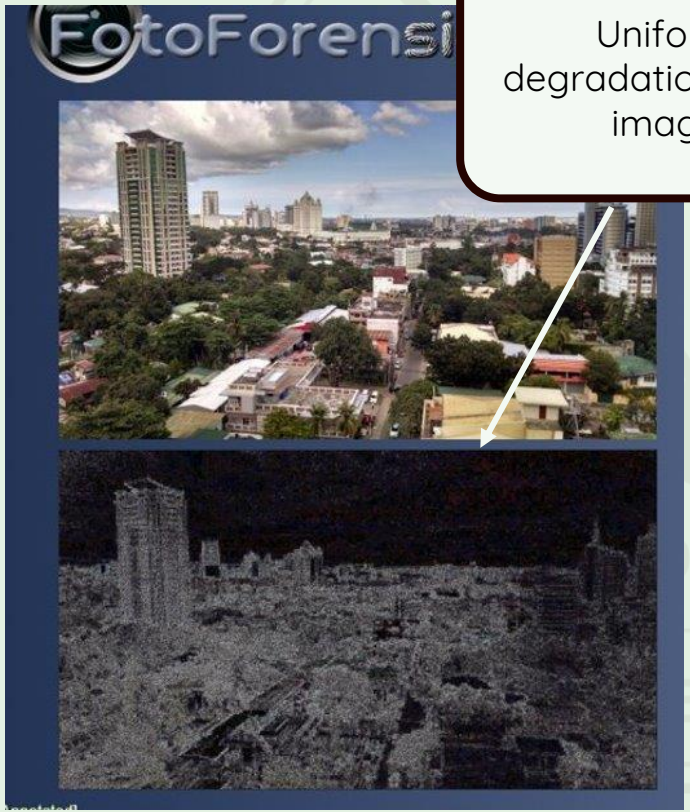
Image: Test Image 9

- **Fotoforensic**

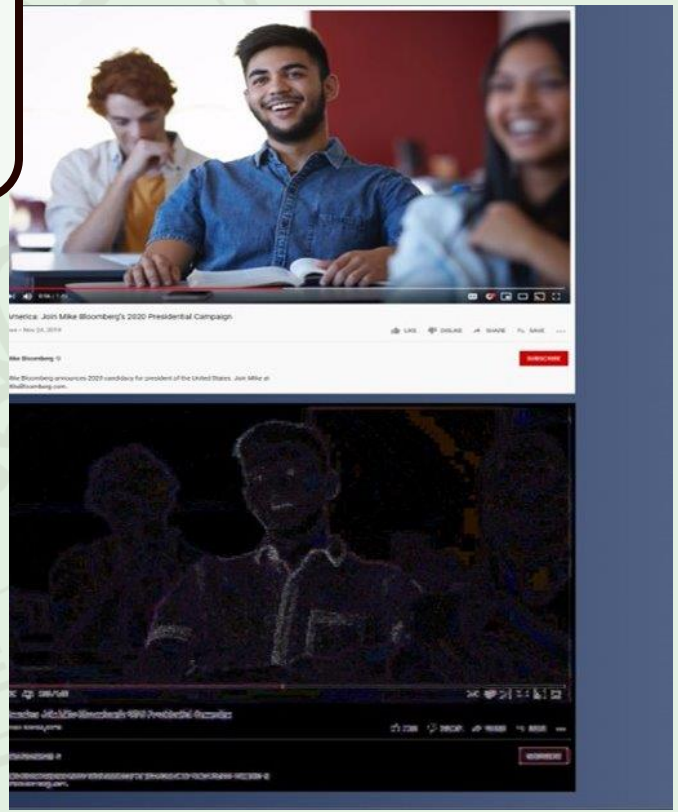




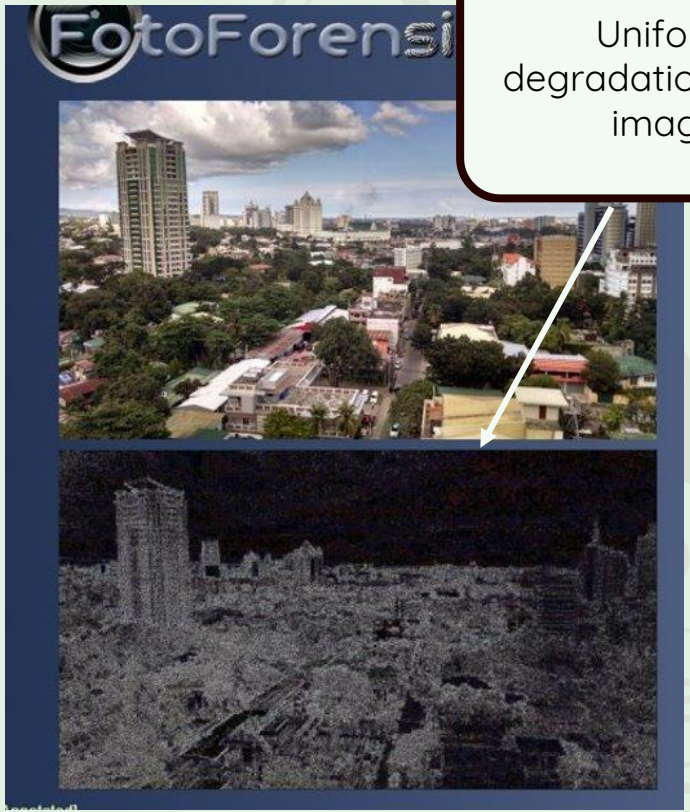
- **Fotoforensic**



Uniform degradation of the image



- **Fotoforensic**



Conclusion

Identified Google as the most effective algorithm followed by Bing and Yandex

Identified specific algorithms used by current state-of-the-art models

Google and Bing perform well on landscape images while Yandex performs well on face images

No engine able to identify fake images or flag them as suspicions

Found variance between different users

Future Work

Continue with more images and a wider variety of images

Investigate how image metadata effects algorithm performance

Investigate if algorithm can identify edited or generated images

- **Thank You For Watching!**

Citations

A. Toler, “Guide to using reverse image search for investigations:, Dec.2019
https://www.bellingcat.com/resources/how-tos/2019/12/26/guide-to-using-reverse-image-search-for-investigations/ (last accessed December 20, 2022)

M. Lew, N. Sebe, C. Djeraba, and R. Jain, “Content-based multimedia information retrieval: state of the art and challenges”, ACM Trans. On Multimedia Computing, Communications, and Applications, Feb. 2006
http://www.ugmode.com/prior_art/lew2006cbm.pdf (last accessed December 20, 2022)

A. Koul, S. Ganju, and M. Kasam, Practical deep learning for mobile, cloud, and edge, Chapter 4, O’Reily, New York, NY
Z.
Wang, Y. Mei, and F. Yan, “A new web image search engine by using SIFT algorithm”, Proc. Int. Conf. on web systems and mining 2009, https://www.computer.org/csdl/proceedingsarticle/wism/2009/3817a366/12OmNzd7bLg (last accessed December 20, 2022)

N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection”,
http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf (last accessed December 20, 2022)

J. Matas et.al., “Robust wide baseline stereo from maximally stable external regions”, Proc. British Machine Vision Conf. p.
384-396 (2002)

H. Hu et.al., “Web scale responsive visual search at Bing”, https://dl.acm.org/doi/pdf/10.1145/3219819.3219843 (last accessed
December 20, 2022)

Rey, L. A. P., Menkovski, V., & Portegies, J. W. (2019). Diffusion Variational Autoencoders. CoRR, abs/1901.08991. Retrieved
from http://arxiv.org/abs/1901.08991

Flowers before difference of gaussians.jpg. (2022, June 3). Wikimedia Commons, the free media repository. Retrieved