Image Processing

Tushar B. Kute,
http://tusharkute.com

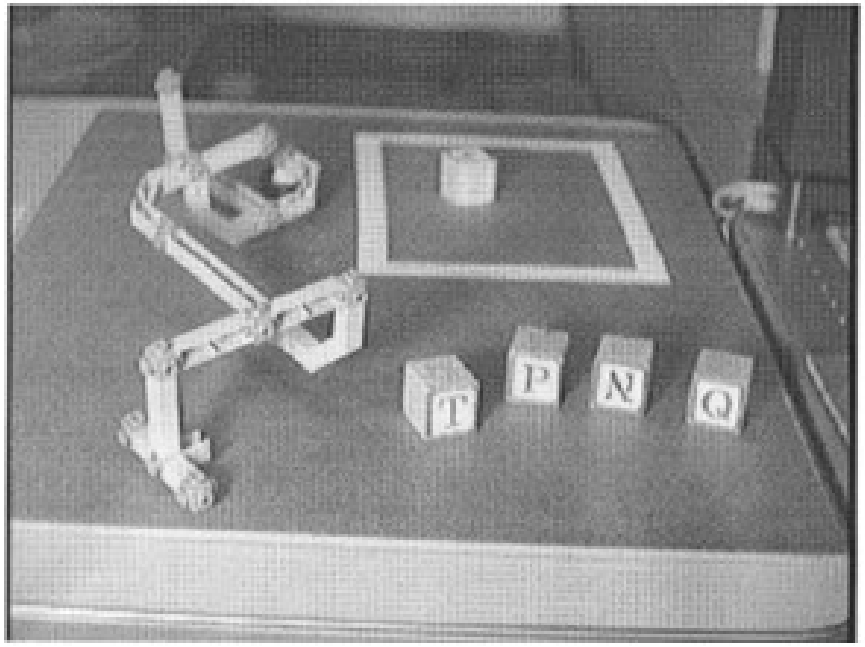
Edge Detection

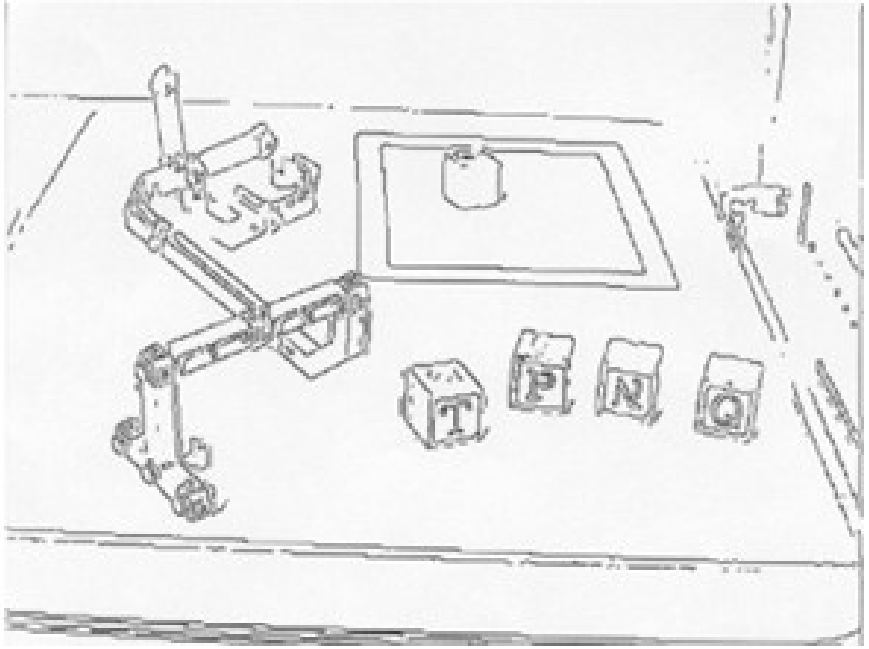
- Edge detection is a technique of image processing used to identify points in a digital image with discontinuities, simply to say, sharp changes in the image brightness.
- These points where the image brightness varies sharply are called the edges (or boundaries) of the image.
- It is one of the basic steps in image processing, pattern recognition in images and computer vision. When we process very high-resolution digital images, convolution techniques come to our rescue.

Edge Detection

- Edge detection is mostly used for the measurement, detection and location changes in an image gray. Edges are the basic feature of an image. In an object, the clearest part is the edges and lines.
- With the help of edges and lines, an object structure is known. That is why extracting the edges is a very important technique in graphics processing and feature extraction.
- The basic idea behind edge detection is as follows:
  - To highlight local edge operator use edge enhancement operator.
  - Define the edge strength and set the edge points.

Edge Detection





Sobel Edge Detection

- The Sobel edge detection operator extracts all the edges of an image, without worrying about the directions.
- The main advantage of the Sobel operator is that it provides differencing and smoothing effect.
- Sobel edge detection operator is implemented as the sum of two directional edges.
- And the resulting image is a unidirectional outline in the original image.

Sobel Edge Detection

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Sobel Edge Detection

- Sobel Edge detection operator consists of 3x3 convolution kernels. Gx is a simple kernel and Gy is rotated by 90°
- These Kernels are applied separately to input image because separate measurements can be produced in each orientation i.e Gx and Gy.

Sobel Edge Detection

- Following is the gradient magnitude:

$$|G| = \sqrt{Gx^2 + Gy^2}$$

- As it is much faster to compute An approximate magnitude is computed:

$$|G| = |Gx| + |Gy|$$

Robert’s Edge Detection

- Robert's cross operator is used to perform 2-D spatial gradient measurement on an image which is simple and quick to compute.
- In Robert's cross operator, at each point pixel values represents the absolute magnitude of the input image at that point.
- Robert's cross operator consists of 2x2 convolution kernels. Gx is a simple kernel and Gy is rotated by 90o

Robert's Edge Detection

+1	0
0	-1

Gx

0	+1
-1	0

Gy

Robert's Edge Detection

- Following is the gradient magnitude:

$$|G| = \sqrt{Gx^2 + Gy^2}$$

- As it is much faster to compute An approximate magnitude is computed:

$$|G| = |Gx| + |Gy|$$

Canny Edge Detection

- The Canny edge detection algorithm is a multi-stage algorithm that is used to detect edges in images.
- It was developed by John F. Canny in 1986. The algorithm consists of the following steps:

Canny Edge Detection

- Convert the image to grayscale
- Reduce noise – as the edge detection that using derivatives is sensitive to noise, we reduce it.
- Calculate the gradient – helps identify the edge intensity and direction.
- Non-maximum suppression – to thin the edges of the image.
- Double threshold – to identify the strong, weak and irrelevant pixels in the images.
- Hysteresis edge tracking – helps convert the weak pixels into strong ones only if they have a strong pixel around them.

Laplacian Edge Detection

- The Laplacian edge detection algorithm is a simple and efficient method for detecting edges in images.
- It works by calculating the Laplacian of the image, which is a measure of the second derivative of the image.
- The Laplacian of an image is zero at all points where the image is smooth, but it will be positive at points where the image has a sharp change in intensity, such as at edges.

Positive Laplacian Operator

- In Positive Laplacian we have standard mask in which center element of the mask should be negative and corner elements of mask should be zero.

0	1	0
1	-4	1
0	1	0

- Positive Laplacian Operator is use to take out outward edges in an image.

Negative Laplacian Operator

- In negative Laplacian operator we also have a standard mask, in which center element should be positive.
- All the elements in the corner should be zero and rest of all the elements in the mask should be -1.

0	-1	0
-1	4	-1
0	-1	0

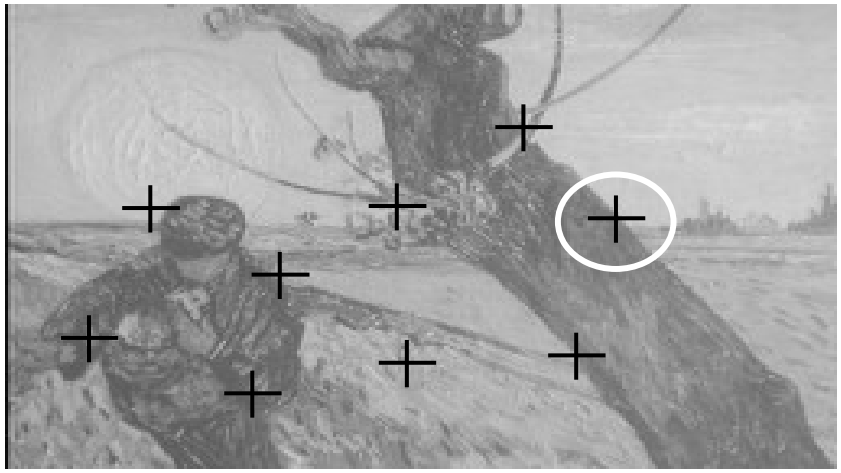
- Negative Laplacian operator is use to take out inward edges in an image

Laplacian Edge Detection

- The Laplacian edge detection algorithm can be implemented using a 3x3 kernel, which is a small square matrix that is used to convolve the image.
- The kernel is typically centered at a pixel that is being considered for edge detection, and the values in the kernel are used to calculate the Laplacian of the image at that pixel.
- If the Laplacian of the image at a pixel is positive, then that pixel is considered to be an edge pixel.
- The Laplacian edge detection algorithm can be used to detect edges in both grayscale and color images.

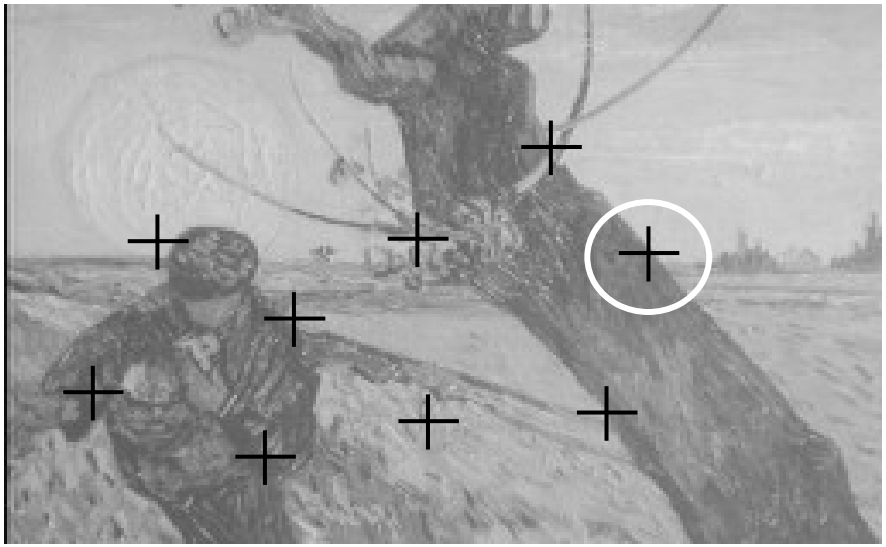
Interest Points

- A point in an image which has a well-defined position and can be robustly detected.
- Typically associated with a significant change of one or more image properties simultaneously (e.g., intensity, color, texture).



Interest Points

- Corners is a special case of interest points.
- However, interest points could be more generic than corners.

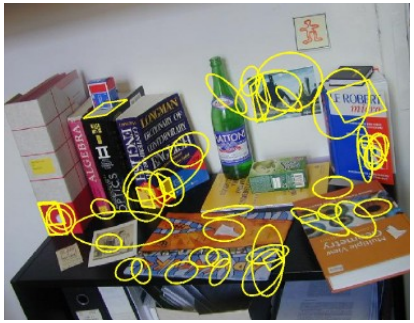


Interest Points: Why useful?

- Could be used to find corresponding points between images which is very useful for numerous applications!

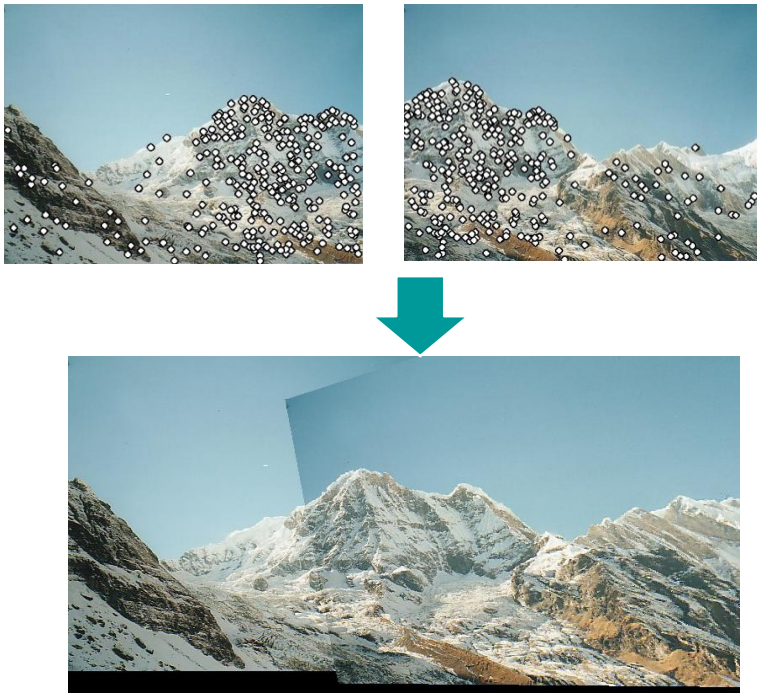
stereo matching

left camera



right camera

panorama stitching

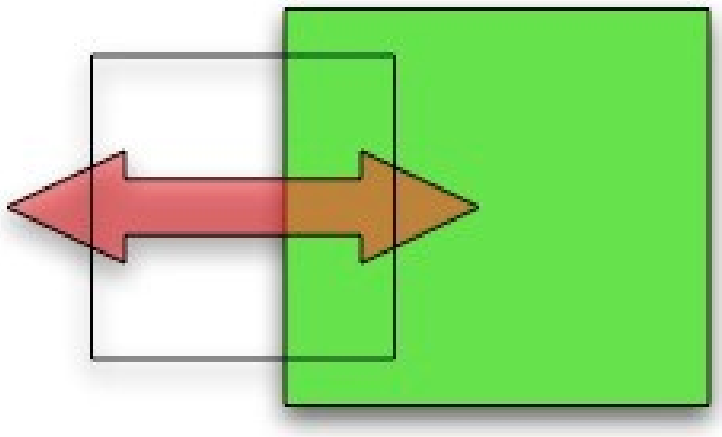


Corner Detection

- Corner detection is very useful for shape description and matching.
- Corner detection works on the principle that if you place a small window over an image, if that window is placed on a corner then if it is moved in any direction there will be a large change in intensity.
- If the window is over a flat area of the image then there will be obviously be no intensity change when the window moves.
- If the window is over an edge there will only be an intesity change if the window moves in one direction.

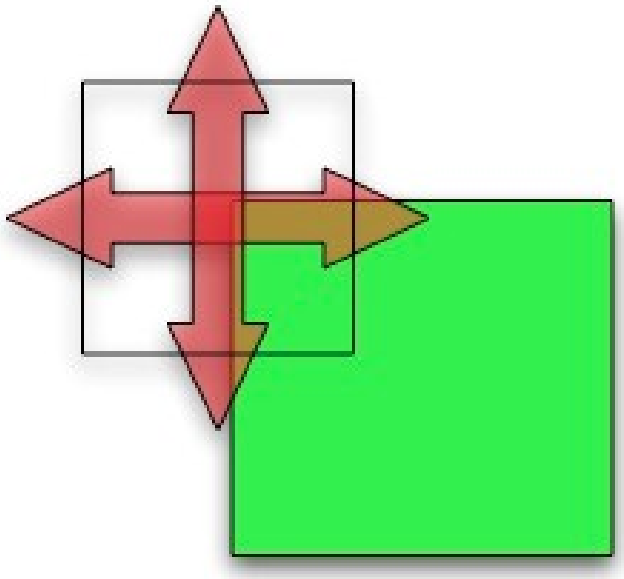
Corner Detection

- If the window is over a corner then there will be a change in all directions, and therefore we know there must be a corner.



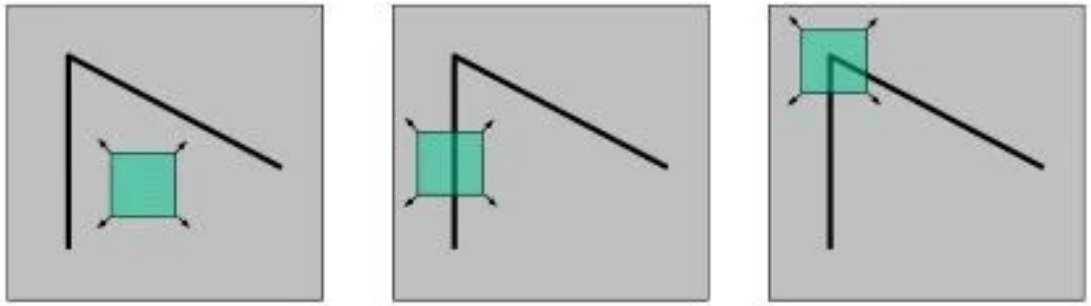
Corner Detection

- The Harris corner detector, measures the strength of detected corners, and only marks those above a given strength as actual corners.
- The number detected can be altered by varying the value of k.



Corner Detection

- Corners are the important features in the image, and they are generally termed as interest points which are invariant to translation, rotation, and illumination.

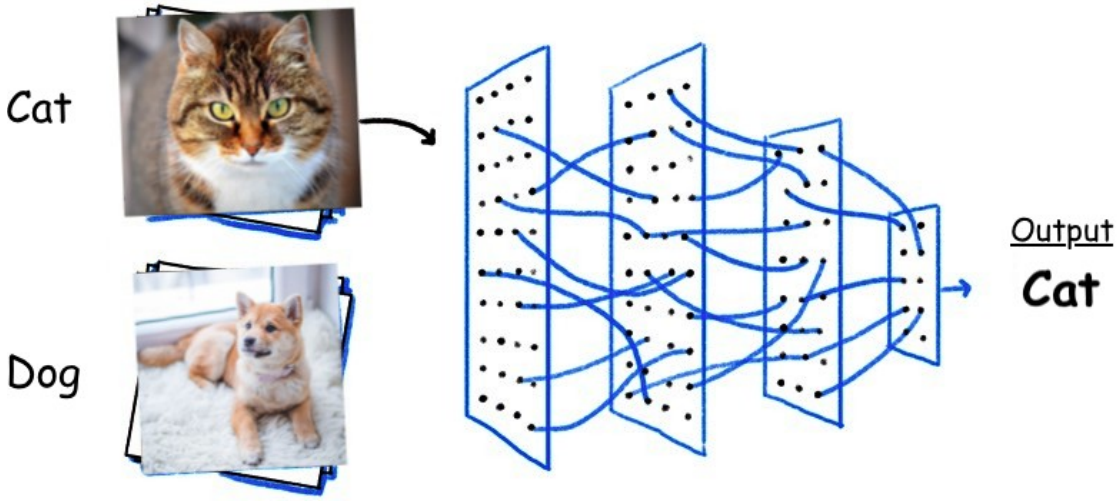


Harris Corner Detection

- 1. Take the grayscale of the original image
- 2. Apply a Gaussian filter to smooth out any noise
- 3. Apply Sobel operator to find the x and y gradient values for every pixel in the grayscale image
- 4. For each pixel p in the grayscale image, consider a 3×3 window around it and compute the corner strength function. Call this its Harris value.
- 5. Find all pixels that exceed a certain threshold and are the local maxima within a certain window (to prevent redundant dupes of features)
- 6. For each pixel that meets the criteria in 5, compute a feature descriptor.

Image Classification

- Image classification is a supervised learning problem: define a set of target classes (objects to identify in images), and train a model to recognize them using labeled example photos.



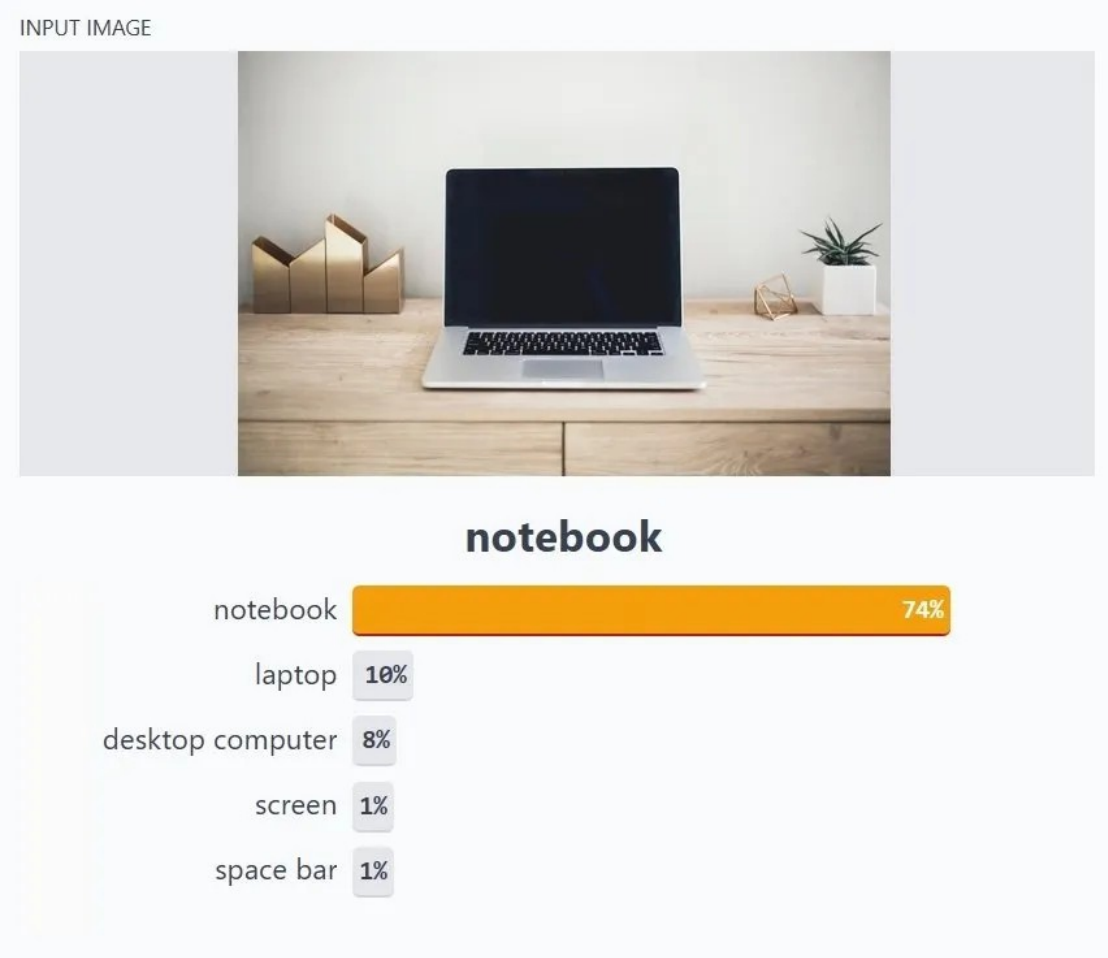
Unsupervised Classification

- Unsupervised classification technique is a fully automated method that does not leverage training data.
- This means machine learning algorithms are used to analyze and cluster unlabeled datasets by discovering hidden patterns or data groups without the need for human intervention.
- With the help of a suitable algorithm, the particular characterizations of an image are recognized systematically during the image processing stage.

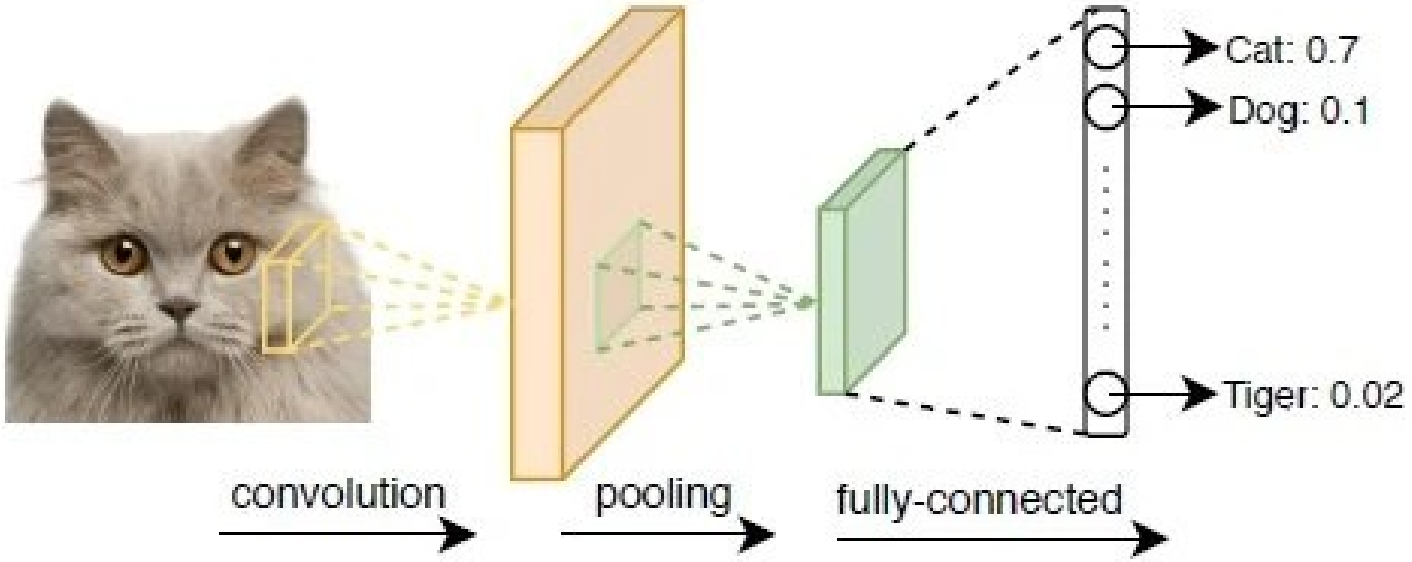
Supervised Classification

- Supervised image classification methods use previously classified reference samples (the ground truth) in order to train the classifier and subsequently classify new, unknown data.
- Therefore, the supervised classification technique is the process of visually choosing samples of training data within the image and allocating them to pre-chosen categories, including vegetation, roads, water resources, and buildings.
- This is done to create statistical measures to be applied to the overall image.

Supervised Classification



Convolutional Neural Network

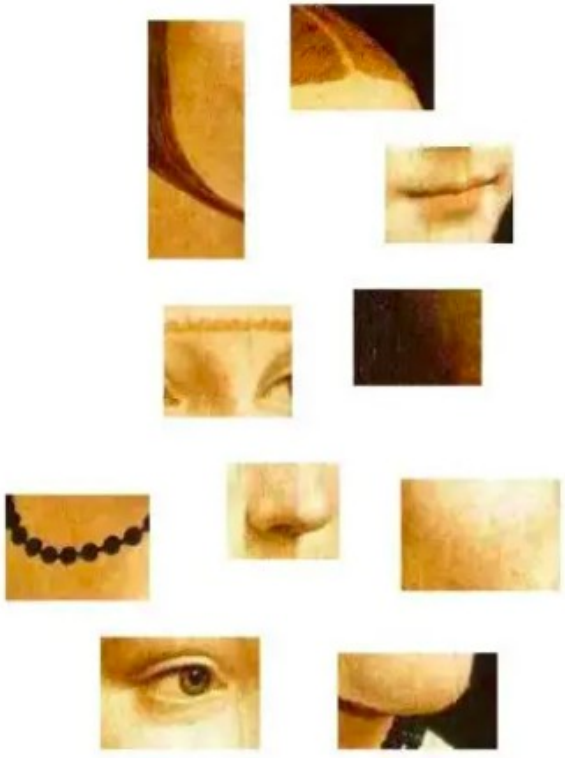


Bag of Features

- Bag of Features (BoF) is a technique used in computer vision and image processing to extract and represent features from images in a compact and meaningful way.
- The basic idea behind BoF is to extract local features from an image, such as SIFT, SURF, or ORB, and then use clustering techniques to group the features into a set of visual words.
- Each image is then represented by a histogram of these visual words, which is called a bag of features.

Bag of Features

Face



features

Bag of Features

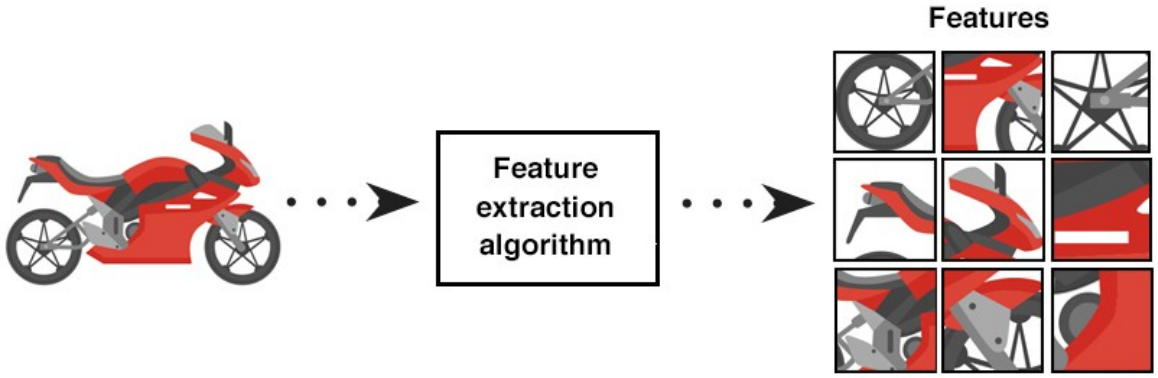
- The Bag of features representation is used in many computer vision and image processing tasks such as image retrieval, object recognition, and semantic segmentation.
- In image retrieval, BoF is used to represent images compactly and efficiently, allowing for fast and accurate retrieval of similar images.
- In object recognition, BoF extracts features from images and trains a classifier to recognize objects in new images.
- In semantic segmentation, BoF is used to extract features from images and train a model to predict the semantic labels of the pixels in the image.

Steps Involved in The BoF Process

- The basic steps involved in the Bag of Features (BoF) method include
  - feature extraction,
  - clustering, and
  - histogram representation.

Feature Extraction

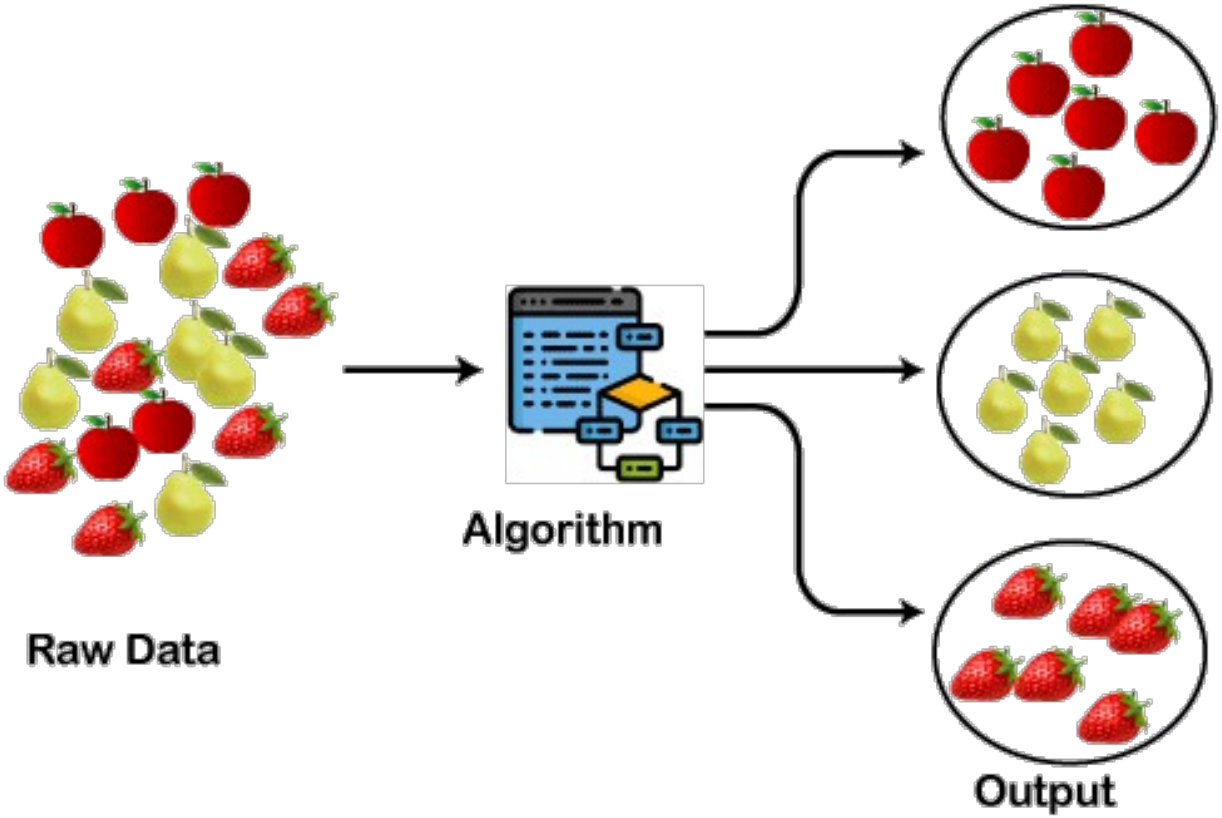
- The first step in the BoF method is to extract local features from the images.
- This is done using feature detection and description methods such as SIFT, SURF, or ORB. These methods extract a set of key points and associated descriptor vectors from the image.



Clustering

- The next step is to group the extracted features into a set of visual words.
- This is done by applying clustering techniques such as k-means or hierarchical clustering to the descriptor vectors.
- The result is a set of clusters, where each cluster represents a visual word.

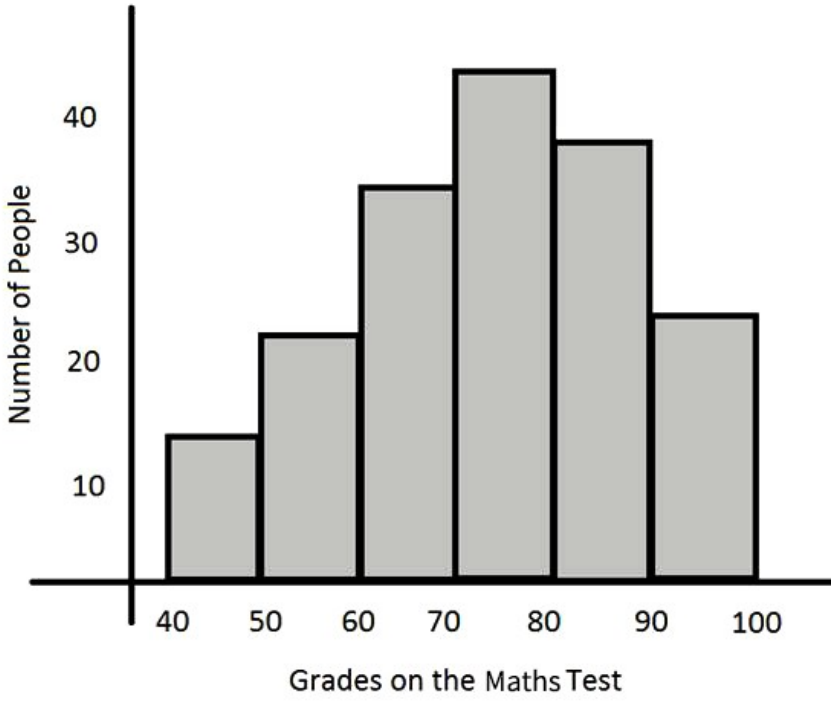
Clustering



Histogram

- Finally, the image is represented by a histogram of the visual words.
- This is done by counting the number of features that belong to each visual word and creating a histogram of these counts.
- The resulting histogram is the bag of features representation of the image.

Histogram


Number of People
40
30
20
10
40
50
60
70
80
90
100
Grades on the Maths Test

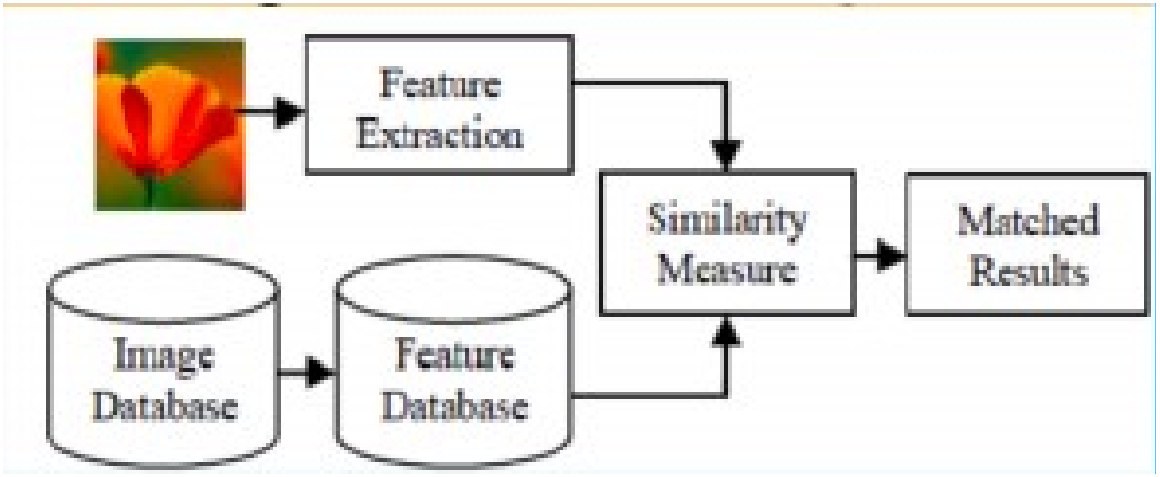
Bag of Features

- Once the feature extraction, clustering, and histogram representation steps are completed, we can use the bag of feature representation of the images to perform various tasks such as image retrieval, object recognition, and semantic segmentation.
-
- It is important to note that the number of clusters, or visual words, used in the BoF method will affect the representation of the images. Using a larger number of clusters will result in a more detailed representation, but it will also increase the computational cost and memory usage.

Bag of Features: Real World Use Cases

- Image retrieval:
  - BoF has been used in image retrieval systems to represent images compactly and efficiently, allowing for fast and accurate retrieval of similar images.
  - It has been used in applications such as image search engines, where users can search for images based on keywords or visual similarity.

Bag of Features: Real World Use Cases



Bag of Features: Real World Use Cases

- Object recognition:
  - BoF has been used to extract features from images and train a classifier to recognize objects in new images.
  - It has been used in applications such as surveillance systems, where it can be used to recognize and track objects in video streams automatically.

Bag of Features: Real World Use Cases

Bag of Features: Real World Use Cases

- Semantic segmentation:
  - BoF has been used to extract features from images and train a model to predict the semantic labels of the pixels in the image.
  - It has been used in applications such as autonomous driving, where it can be used to segment the road, vehicles, pedestrians, and other objects in an image.

Bag of Features: Real World Use Cases

predict

Person
Bicycle
Background

Bag of Features: Real World Use Cases

- Medical imaging:
  - BoF has been used in medical imaging to segment and classify the structures in medical images such as CT and MRI scans.
  - It allows for the automated detection and diagnosis of diseases, making it a valuable tool for radiologists and physicians.
- Robotics:
  - BoF has been used in robotics for object recognition and localization.
  - It allows robots to understand their environment and identify objects, allowing them to interact with the environment more effectively.

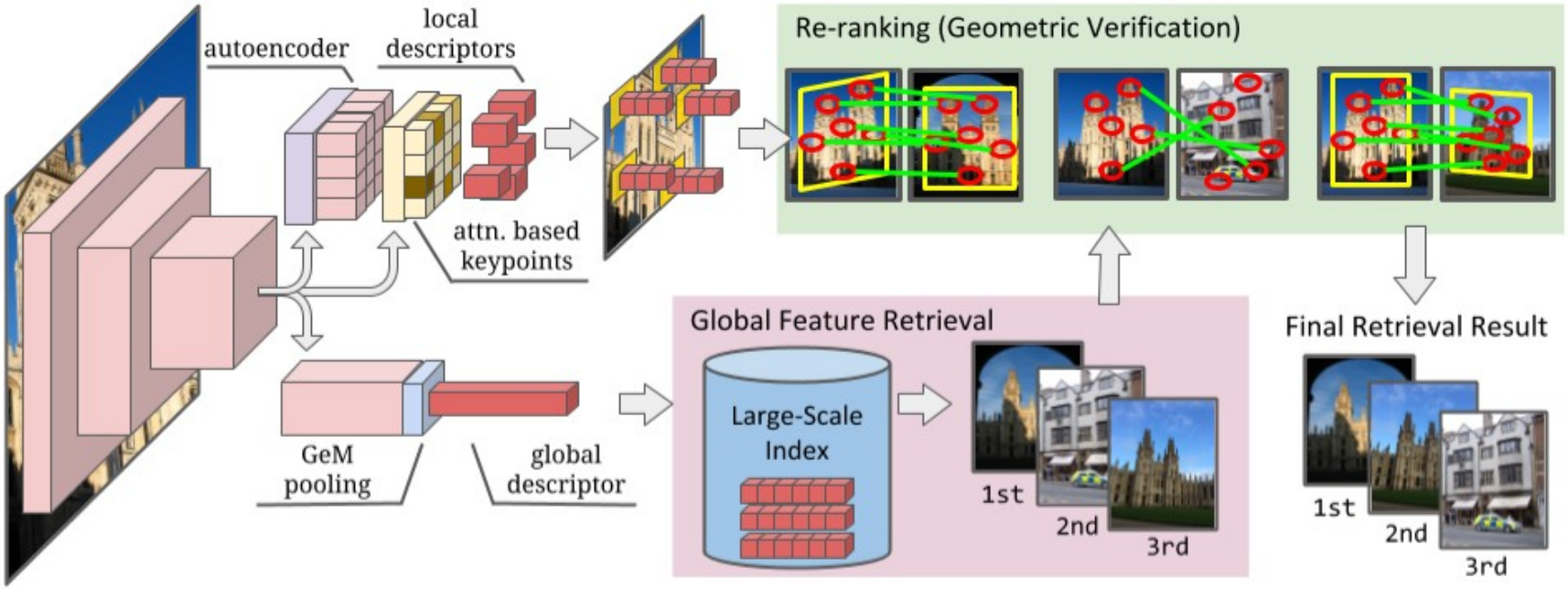
Large-scale Instance Recognition

- Large-scale instance recognition (LIR) is a computer vision task that aims to recognize a specific instance of an object in a large dataset of images.
- This is a challenging task because it requires the model to be able to identify the same object even when it appears in different poses, sizes, and illumination conditions.

Large-scale Instance Recognition

- LIR has a wide range of applications, such as:
  - Image retrieval: LIR can be used to retrieve images of a specific object from a large dataset. For example, you could use LIR to find all the images of the Eiffel Tower in a collection of millions of images.
  - Object tracking: LIR can be used to track the movement of an object in a video. This can be useful for applications such as surveillance or robotics.
  - Scene understanding: LIR can be used to understand the context of an image. For example, you could use LIR to determine that an image of a car is in a parking lot.

Large-scale Instance Recognition



Large-scale Instance Recognition

- There are a number of challenges that need to be addressed in order to achieve good performance in LIR. These challenges include:
  - Object variability: The same object can appear in a wide variety of poses, sizes, and illumination conditions. The model needs to be able to handle this variability in order to recognize the object correctly.

Large-scale Instance Recognition

- Data scarcity:
  - There is often not enough data available to train a model for LIR.
  - This is because it is difficult to collect a large dataset of images of the same object in different poses, sizes, and illumination conditions.
- Computational complexity:
  - LIR is a computationally expensive task.
  - This is because the model needs to be able to process a large number of images in order to find the instances of the target object.

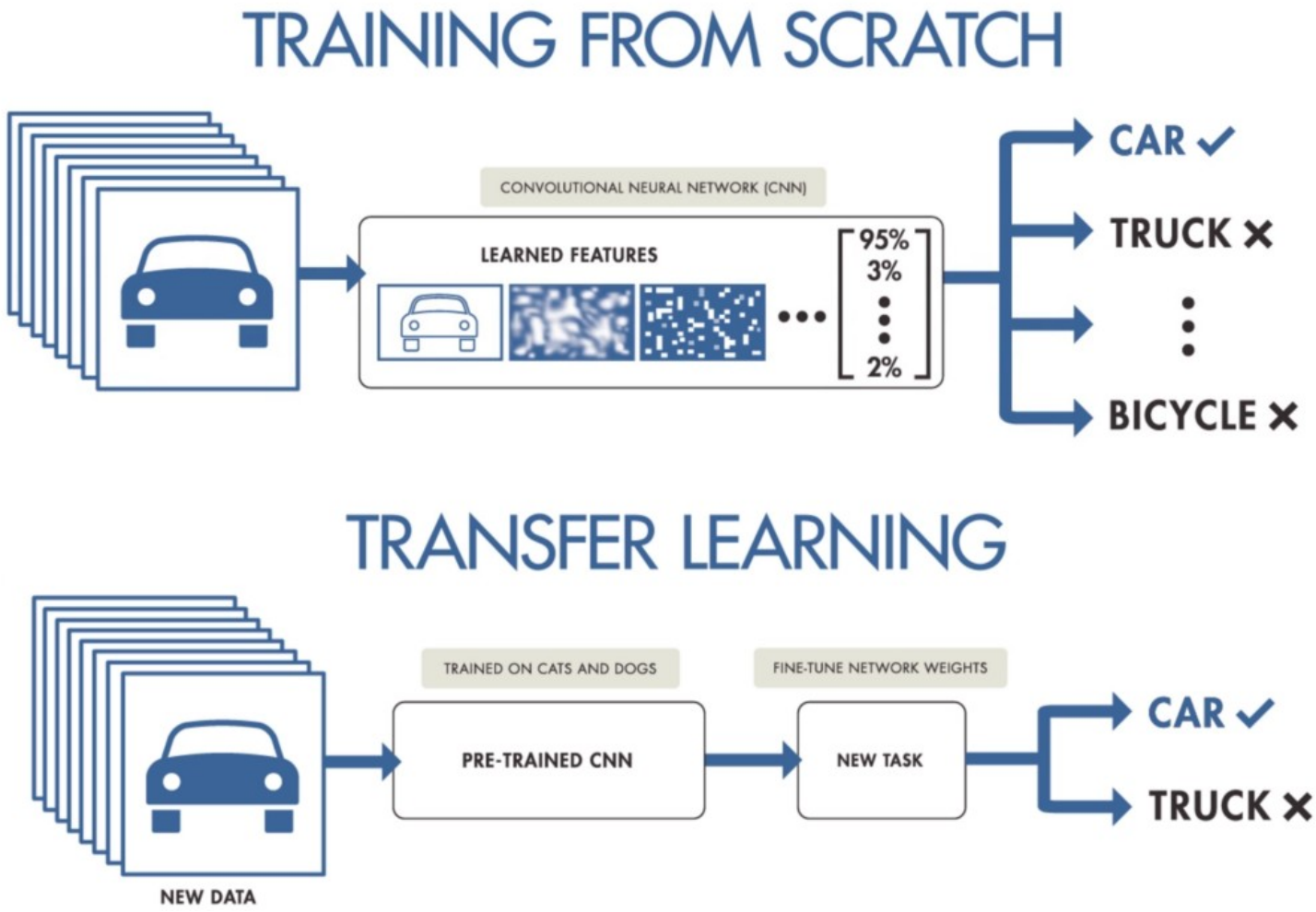
Transfer Learning

- Transfer learning is a machine learning technique where a model trained on one task is reused as the starting point for a model on a second task.
- This can be done by freezing the weights of the first model and then training the second model on a new dataset.
- This can help to improve the performance of the second model, as it can start from a point where the first model has already learned some important features.

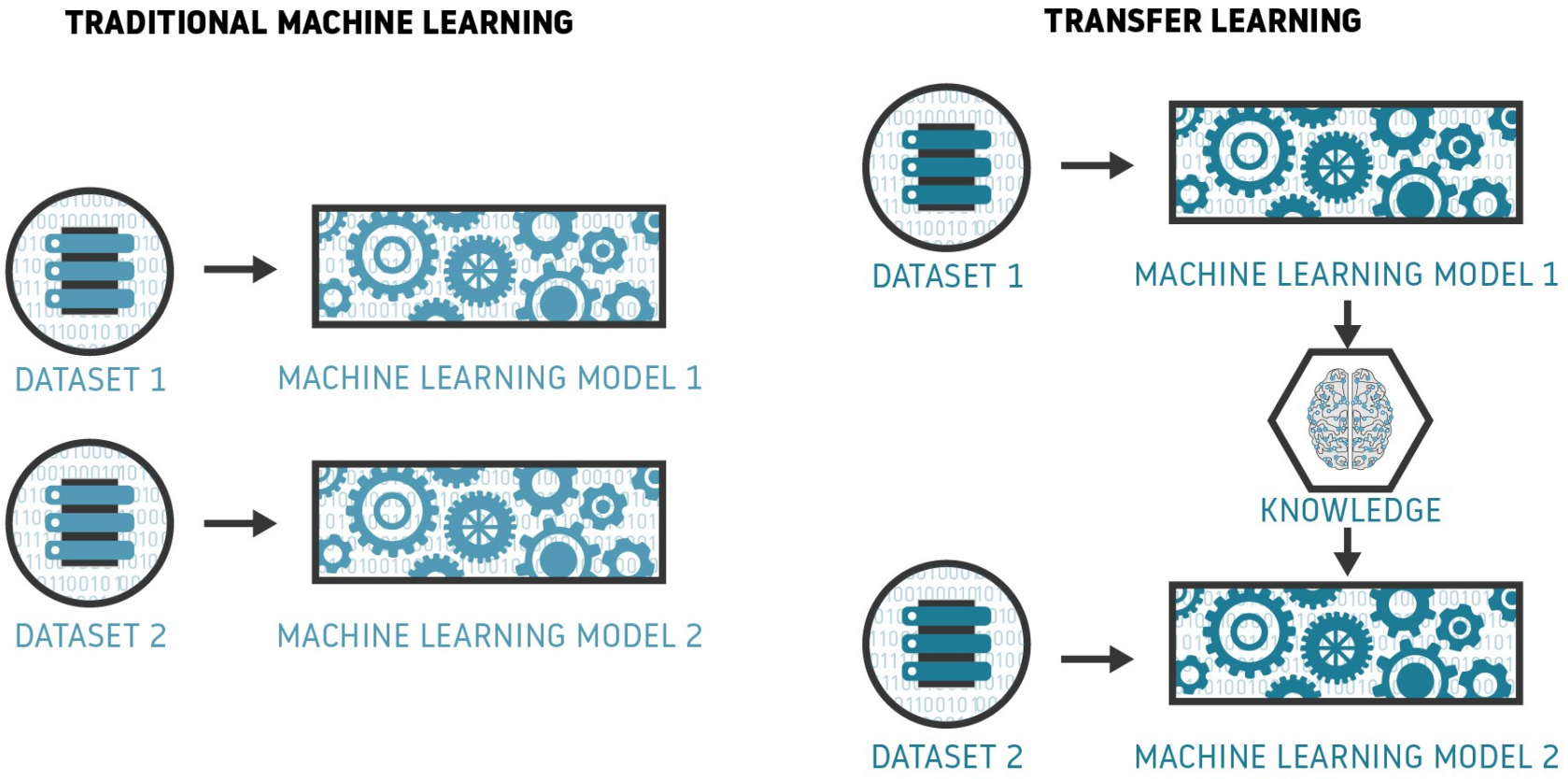
Transfer Learning

- Transfer learning is a popular approach in deep learning, as it can help to reduce the amount of data that needs to be collected for the second task.
- This can be especially useful when the second task has a small dataset, as it can be difficult to train a model from scratch on such a small dataset.

Transfer Learning



Transfer Learning



Transfer Learning: Benefits

- Reduces the amount of data needed: Transfer learning can help to reduce the amount of data needed to train a model. This can be especially useful when the second task has a small dataset.
- Improves performance: Transfer learning can help to improve the performance of a model. This is because the first model has already learned some important features, which can be reused by the second model.
- Saves time: Transfer learning can save time, as it does not require the second model to be trained from scratch.

Transfer Learning: Limitations

- The first model must be trained on a related task: The first model must be trained on a task that is related to the second task. This is because the first model will learn features that are relevant to the second task.
- The second model may not generalize well: The second model may not generalize well to new data. This is because the second model is only trained on a subset of the possible data.
- The first model may not be available: The first model may not be available. This is because the first model may have been trained by someone else, or it may not be publicly available.

Transfer Learning: Develop Model Approach

- 1. Select Source Task. You must select a related predictive modeling problem with an abundance of data where there is some relationship in the input data, output data, and/or concepts learned during the mapping from input to output data.
- 2. Develop Source Model. Next, you must develop a skillful model for this first task. The model must be better than a naive model to ensure that some feature learning has been performed.

Transfer Learning: Develop Model Approach

- 3. Reuse Model. The model fit on the source task can then be used as the starting point for a model on the second task of interest. This may involve using all or parts of the model, depending on the modeling technique used.
- 4. Tune Model. Optionally, the model may need to be adapted or refined on the input-output pair data available for the task of interest.

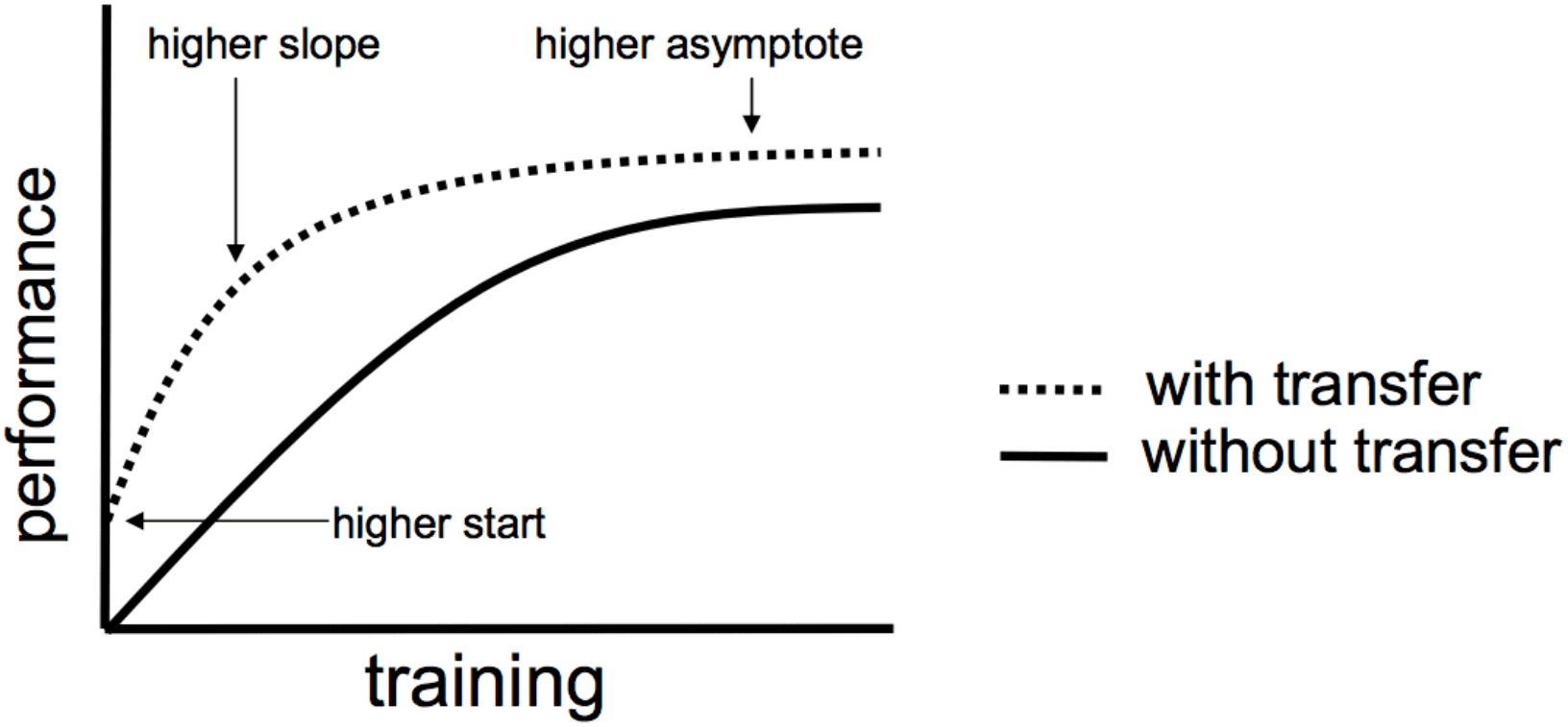
Pre-trained Model Approach

- Select Source Model. A pre-trained source model is chosen from available models. Many research institutions release models on large and challenging datasets that may be included in the pool of candidate models from which to choose from.
- Reuse Model. The model pre-trained model can then be used as the starting point for a model on the second task of interest. This may involve using all or parts of the model, depending on the modeling technique used.
- Tune Model. Optionally, the model may need to be adapted or refined on the input-output pair data available for the task of interest.

Transfer Learning: Why?

- Higher start. The initial skill (before refining the model) on the source model is higher than it otherwise would be.
- Higher slope. The rate of improvement of skill during training of the source model is steeper than it otherwise would be.
- Higher asymptote. The converged skill of the trained model is better than it otherwise would be.

Transfer Learning: Why?



Thank you

@mitu_skillologies

@mITuSkillologies

@mitu_group

@mitu-skillologies

@MITUSkillologies

kaggle

@mituskillologies

Web Resources
https://mitu.co.in
http://tusharkute.com

@mituskillologies

contact@mitu.co.in

tushar@tusharkute.com