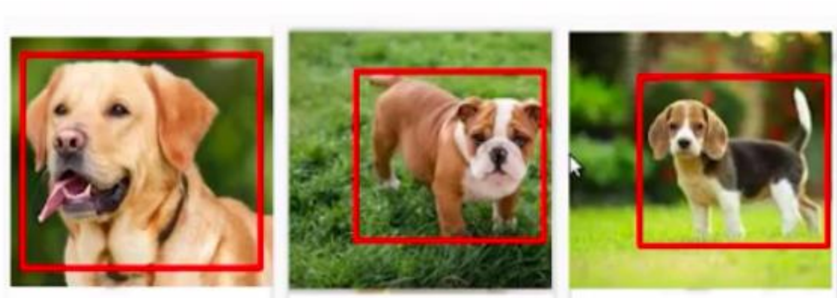
UNIT - V: Applications

Applications: Object recognition, sparse coding, computer vision, natural language processing. Introduction to Deep Learning Tools: Tensor Flow, Caffe, Theano, Torch.

Object Recognition

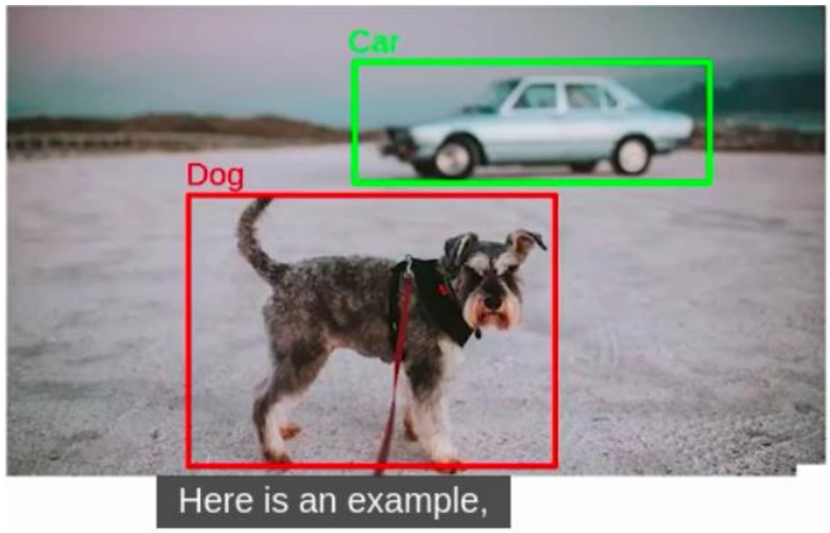
- Object detection is a computer vision technique for locating instances of objects in images or videos.
- Humans can easily detect and identify objects present in an image.

- So instead of classifying, which type of dog is present in these images, we have to actually **locate a dog** in the image.
- That is, I have to find out where is the dog present in the image? Is it **at the center or at the bottom left**? And so on.
- Well, we can **create a box** around the dog that is present in the image and specify the x and y coordinates of this box.



- for now, consider that the location of the object in the image can be represented as coordinates of these boxes.

- So this box around the object in the image is formally known as a **bounding box**.
- Now, this becomes an image localization problem where we are given a set of images and we have to identify where is the object present in the image.



- In the case of object detection problems, we have to classify the objects in the image and also locate where these objects are present in the image.
- But the image classification problem had only one task where we had to classify the objects in the image.
- So broadly we have three tasks for object detection problems:

  1. To identify if there is an object present in the image,
  2. where is this object located,
  3. what is this object?

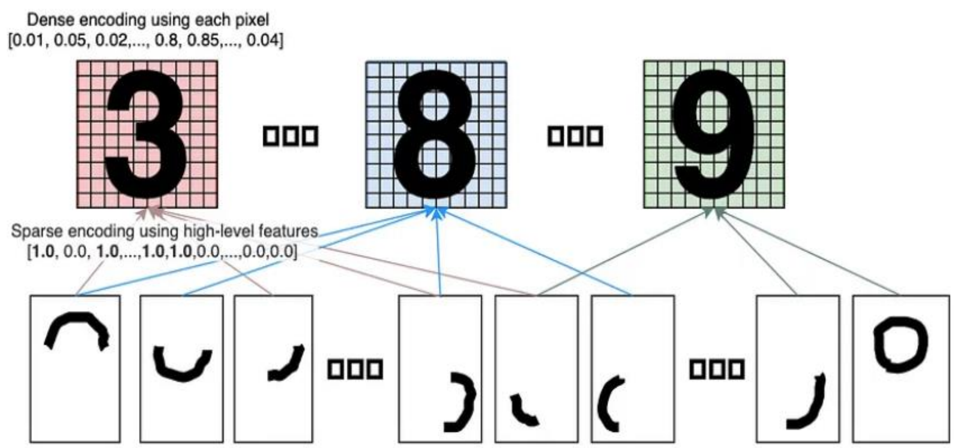
- Now the object detection problem can also be divided into multiple categories.
- First is the case when you have images that have **only one object.**
- That is you can have 1000 images in the data set, and all of these images will have only one object. And if all these objects belong to a **single class**
- Another problem could be where you are provided with multiple images, and within each of these images, you have **multiple objects.**

- Also, these objects can be of the **same class,** or another problem can be that these objects are of **different classes.**
- So in case you have multiple objects in the image and all of the objects are of different classes.
- you would have to not only locate the objects but also classify these objects.

Sparse coding

- sparse coding is a powerful machine learning technique that allows us to find compact, efficient representations of complex data.
- It has been widely applied to various applications such as brain imaging, natural language processing, and image and sound processing.
- *Sparse coding* is an unsupervised type of representation learning.
- It aims to find a sparse representation of a given data.
- In other words for our input (i.e. dataset) we try to find another representation.
- If we treat our dataset item (i.e. image) as a vector of some dimensionality, those vectors form some distribution in that space.
- By learning new representation, we are looking for another distribution, more useful one.
- The two main objectives are, there should be a linear function that translates any image from learned representation to original one
- new representation should be *sparse*
- Sparse means most of the values of the vector in the new representation will be zeros.
- In other words single item (i.e. image) from the dataset will be represented by few features out of our learned dictionary of features (factors).
- Take a MNIST image, handwritten b/w digit: it's original input is 28x28 scalar values.
- Vector of these values represent s intensity of every pixel, and is quite dense: for each image high percentage of pixels are not equal to zero.
- With natural images almost all pixels are not close to 0.

- If we try to sample such image given this representation, we will end up getting a **random noise**, because we actually don't know underlying distribution that produced this dataset.
- But if we learn sparse representation of our dataset, where each image (handwritten digit from 0 to 9) is represented(not zero value) only by few features, it can actually become useful, because now we know what distribution those features come from.
- For instance, classification models can benefit from such input.

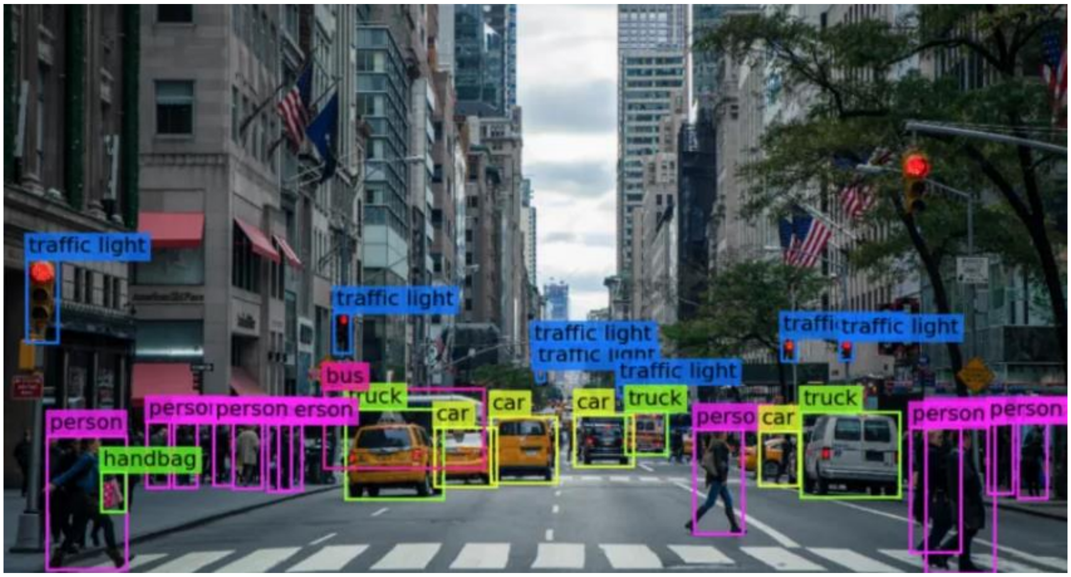


Top: observed images. Bottom: learned set of features, that can be used to sparsely encode observed dataset.

- As an optimization problem we need to find such factors, that with a sparsity constraint can precisely model our original dataset.
- we also need to learn representation coefficients for each sample of our dataset to minimize the difference between modeled sample and original one.

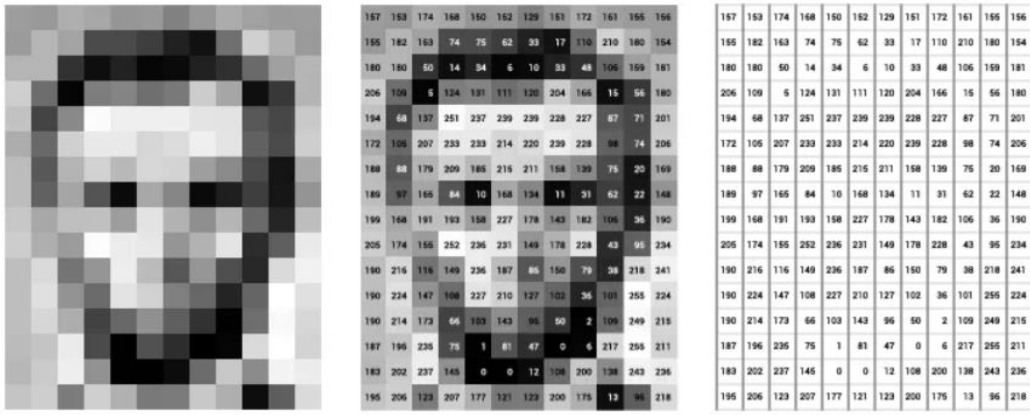
Computer vision

- Computer vision is the field of computer science that focuses on replicating parts of the complexity of the human vision system and enabling computers to identify and process **objects in images and videos** in the same way that humans do.
- One of the driving factors behind the growth of computer vision is the **amount of data** we generate today that is then used to **train and make computer** vision better.

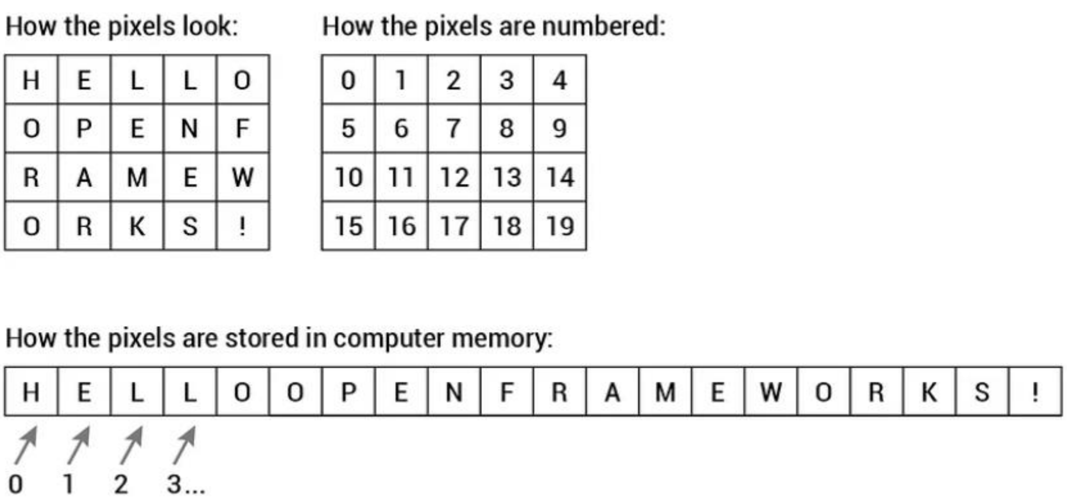


Multi-Object Detection And Classification

- Computer vision is all about pattern recognition.
- So one way to train a computer how to understand visual data is to feed it images, lots of images thousands, millions if possible that have been labeled, and then subject those to various software techniques, or algorithms, that allow the computer to hunt down patterns in all the elements that relate to those labels.
- So, for example, if you feed a computer a **million images of cats,** it will subject them all to algorithms that let them analyze the colors in the photo, the shapes, the distances between the shapes, where objects border each other, and so on,
- so that it identifies a profile of what "**cat**" means.
- When it's finished, the computer will (in theory) be able to use its experience if fed other unlabeled images to find the ones that are of cat.



```
{157, 153, 174, 168, 150, 152, 129, 151, 172, 161, 155, 156,
 155, 182, 163,  74,  75,  62,  33,  17, 110, 210, 180, 154,
 180, 180,  50,  14,  34,   6,  10,  33,  48, 106, 159, 181,
 206, 109,   5, 124, 131, 111, 120, 204, 166,  15,  56, 180,
 194,  68, 137, 251, 237, 239, 239, 228, 227,  87,  71, 201,
 172, 105, 207, 233, 233, 214, 220, 239, 228,  98,  74, 206,
 188,  88, 179, 209, 185, 215, 211, 158, 139,  75,  20, 169,
 189,  97, 165,  84,  10, 168, 134,  11,  31,  62,  22, 148,
 199, 168, 191, 193, 158, 227, 178, 143, 182, 106,  36, 190,
 205, 174, 155, 252, 236, 231, 149, 178, 228,  43,  95, 234,
 190, 216, 116, 149, 236, 187,  86, 150,  79,  38, 218, 241,
 190, 224, 147, 108, 227, 210, 127, 102,  36, 101, 255, 224,
 190, 214, 173,  66, 103, 143,  96,  50,   2, 109, 249, 215,
 187, 196, 235,  75,   1,  81,  47,   0,   6, 217, 255, 211,
 183, 202, 237, 145,   0,   0,  12, 108, 200, 138, 243, 236,
 195, 206, 123, 207, 177, 121, 123, 200, 175,  13,  96, 218};
```



Applications Of Computer Vision

CV In Self-Driving Cars

- Computer vision enables self-driving cars to make sense of their surroundings.
- Cameras capture video from different angles around the car and feed it to computer vision software, which then processes the images in real-time to find the extremities of roads, read traffic signs, detect other cars, objects and pedestrians.
- The self-driving car can then steer its way on streets and highways, avoid hitting obstacles, and (hopefully) safely drive its passengers to their destination.

CV In Facial Recognition

- Computer vision also plays an important role in facial recognition applications, the technology that enables computers to match images of people's faces to their identities.
- Computer vision algorithms detect facial features in images and compare them with databases of face profiles.

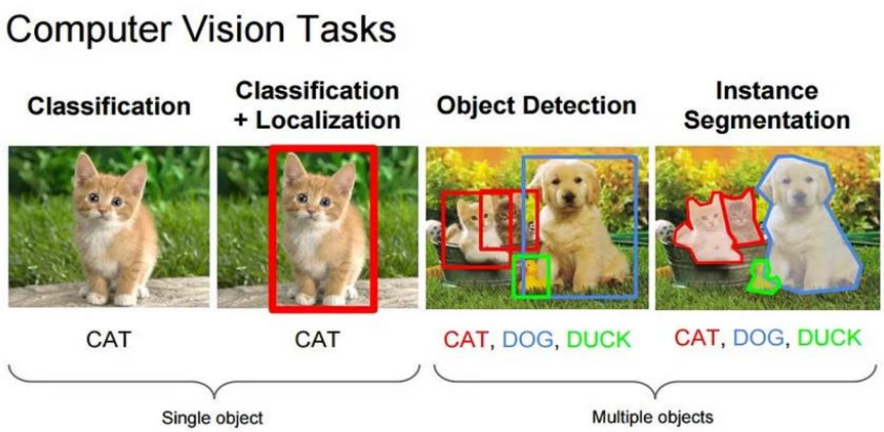
CV In Augmented Reality & Mixed Reality

- Computer vision also plays an important role in augmented and mixed reality, the technology that enables computing devices such as smartphones, tablets and smart glasses to overlay and embed virtual objects on real world imagery.
- Using computer vision, AR gear detect objects in real world in order to determine the locations on a device's display to place a virtual object.

CV In Healthcare

- Computer vision has also been an important part of advances in health-tech. Computer vision algorithms can help automate tasks such as detecting cancerous moles in skin images or finding symptoms in x-ray and MRI scans.

Challenges of Computer Vision



- Many popular computer vision applications involve trying to recognize things in photographs;

for example:

- **Object Classification**: What broad category of object is in this photograph?
- **Object Identification**: Which type of a given object is in this photograph?
- **Object Verification**: Is the object in the photograph?
- **Object Detection**: Where are the objects in the photograph?
- **Object Landmark Detection**: What are the key points for the object in the photograph?
- **Object Segmentation**: What pixels belong to the object in the image?
- **Object Recognition**: What objects are in this photograph and where are they?

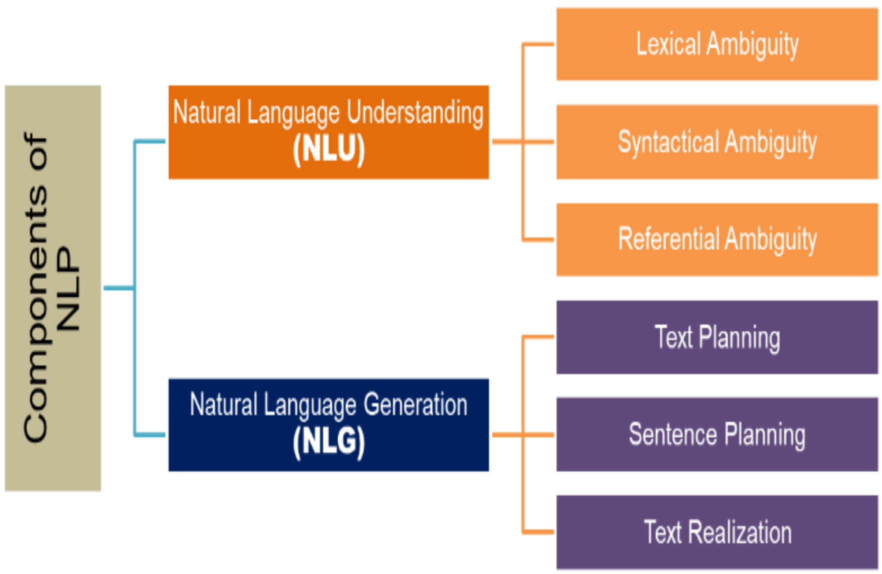
Natural language processing

- Natural Language Processing is the art of extracting information from unstructured text.
- Natural Language Processing (NLP) is basically how you can teach machines to understand **human languages** and extract **meaning from text**.
- NLP is a subfield of Computer Science and Artificial Intelligence that deals with interactions between **computers** and human (natural) **languages.**
- **For Example,** we can use NLP to create AI systems such as,
- Speech Recognition,

- Document Summarization,
- Machine Translation,
- Spam Detection,
- Question Answering,

Components of NLP

- The two basic components in which NLP can be divided are as follows:
- Natural Language Understanding (NLU)
- Natural Language Generation (NLG)



Natural Language Understanding (NLU)

- While learning or trying to interpret a language, there are a lot of ambiguities.
- **Lexical Ambiguity** can occur when a word carries a different sense, i.e. having more than one meaning, and the sentence in which that word is used can be interpreted differently based on its correct sense.
- **Ex::Sentence: The chicken is ready to eat.**
- Is the chicken ready to eat its food or the chicken is ready for someone else to it

- **Syntactical Ambiguity** occurs when we observed that there can be more than one meaning in a **sequence of words**. It is also known as Grammatical ambiguity.
- **Sentence: Chirag met Kshitiz and Dinesh. They went to a restaurant.**
- Here, they refer to Kshitiz and Dinesh or all
- **Referential Ambiguity:** It is very often in a text that it mentions an entity (something/someone), and then refers to it again, possibly in a different sentence, with the help of another word.

Natural Language Generation (NLG)

- It is defined as the process of generating or extracting some meaningful phrases and sentences in the form of natural language with the help of some internal representation.
- This component involves the three basic steps:
- **Text planning:** It involves the retrieving of the relevant information from the knowledge base.
- **Sentence planning:** It involves processes such as choosing required words, forming meaningful phrases, setting the tone of the sentence.
- **Text Realization:** It involves the mapping of sentence plans into sentence structure.

Introduction to Deep Learning Tools

- Deep learning is a subset of machine learning in AI that contains networks that are equipped to learn from unsupervised data.
- It is inspired by the functioning of the human brain for processing information and decision-making purposes.
- Also, it is popularly known as deep neural learning or deep neural network.
- It makes use of algorithms that help in development of models that are capable of predicting better outcomes, which helps the decision making process for any business.
- Deep learning applications are used in different types of industries. One of the examples is: Automated Driving.

- Research is in progress that makes use of deep learning to detect pedestrians, signs, and traffic lights.
- This will not only help in leveraging the power of artificial intelligence but also ensure that there are lesser road accidents.

TensorFlow

- TensorFlow is a deep learning tool that was written in highly-optimized C++ and CUDA (Nvidia's language for programming GPUs) and provides an interface to languages like Python, Java, Go.
- It is an open-source library that was developed by the tech giant, Google, for the smooth running of deep learning applications.
- TensorFlow makes it fairly easy for beginners and even experts to create machine learning models for mobile, web, desktop, and cloud.
- It is also used to create large-scale neural networks with multiple layers.
- If you want to solve deep learning or machine learning problems like Classification, Perception, Understanding, Discovering, Prediction and Creation.

Caffe

- Caffe is a deep learning tool whose framework is built with expression, speed and modularity in consideration.
- Developed by the Berkeley AI Research (BAIR)/The Berkeley Vision and Learning Center (BVLC) and community contributors, it is also an open source deep learning tool.
- Its expressive architecture allows for application and innovation to propagate.
- Caffe is used in academic research, startup prototypes and large-scale industrial applications in speech, vision, and multimedia.
- Caffe has more performance than TensorFlow by 1.2 to 5 times, based on the internal benchmarking in Facebook.

Torch

- Torch is an efficient open source program that uses a LuaJIT scripting language and C/CUDA implementation.

- If you are using this deep learning tool, you will be able to take advantage of its powerful features like: multiple routines for indexing, transposing, slicing, amazing interface to C via LuaJIT, neural network.
- It offers fast, efficient GPU support and is easily embeddable that makes it easier to work with iOS, Android.
- **Theano** is a Python library that is used to evaluate mathematical operations including multi-dimensional arrays.
- It is mainly used in building deep learning projects and is way faster on a GPU than a CPU.
- With its use, it is possible to attain speed that rivals self-made C implementations for problems that involve large amounts of data.