**Unit – V**

|  |  |
| --- | --- |
| **Unit – V**  **Deep Learning Medical Image Classification, Analysis and Visualization** | **Contact Hours = 8 Hours**  **Flipped Classes Content = 2 Hours** |
| Features, Features reduction using Principal Component Analysis (PCA), feature reduction using Image Transforms (DWT), Pre trained CNN Model for feature extraction (only ResNet -50), Example and demonstration of CNN pretrained model for image classification and Identification – Covid-19 Diseases detection using Computed tomography (CT) imageries. | |
| **Topics for Flipped Classes:** Case study review on Pre trained CNN Model | |

**Image Features:**

**Feature (computer vision/Machine Learning/Deep Learning)**

In [image processing](https://en.wikipedia.org/wiki/Image_processing) and computer vision, a feature is a piece of information about the content of an image; typically, about whether a certain region of the image has certain properties. **Features may be specific structures in the image such as points, edges or objects**. Features may also be the result of a **general** [**neighborhood operation**](https://en.wikipedia.org/wiki/Neighborhood_operation) or **feature detection** applied to the image. Other examples of features are related to motion in image sequences, or to shapes defined in terms of curves or boundaries between different image regions.

More broadly a feature is any piece of information which is relevant for solving the computational task related to a certain application. This is the same sense as [feature](https://en.wikipedia.org/wiki/Feature_(machine_learning)) in [machine learning](https://en.wikipedia.org/wiki/Machine_learning) and [pattern recognition](https://en.wikipedia.org/wiki/Pattern_recognition) generally, though image processing has a very sophisticated collection of features. The feature concept is very general and the choice of features in a particular computer vision system may be highly dependent on the specific problem at hand.

## Definition

There is no universal or exact definition of what constitutes a **feature**, and the exact definition often depends on the problem or the type of application. Nevertheless, a feature is typically defined as an "**interesting**" part of an [image](https://en.wikipedia.org/wiki/Digital_image), and features are used as a starting point for many computer vision algorithms.

Since features are used as the starting point and main primitives for subsequent algorithms, the overall algorithm will often only be as good as its feature detector. Consequently, the desirable property for a feature detector is [repeatability](https://en.wikipedia.org/wiki/Repeatability): whether or not the same feature will be detected in two or more different images of the same scene.

Feature detection is a low-level [image processing](https://en.wikipedia.org/wiki/Image_processing) operation. That is, it is usually performed as the first operation on an image, and examines every [pixel](https://en.wikipedia.org/wiki/Pixel) to see if there is a feature present at that pixel. If this is part of a larger algorithm, then the algorithm will typically only examine the image in the region of the features. As a built-in pre-requisite to feature detection, the input image is usually smoothed by a [Gaussian](https://en.wikipedia.org/wiki/Gaussian_blur) kernel in a [scale-space representation](https://en.wikipedia.org/wiki/Scale_space) and one or several feature images are computed, often expressed in terms of local [image derivative](https://en.wikipedia.org/wiki/Image_derivative) operations.

Occasionally, when feature detection is [computationally expensive](https://en.wikipedia.org/wiki/Computationally_expensive) and there are time constraints, a higher level algorithm may be used to guide the feature detection stage, so that only certain parts of the image are searched for features.

There are many computer vision algorithms that use feature detection as the initial step, so as a result, a very large number of feature detectors have been developed. These vary widely in the kinds of feature detected, the computational complexity and the repeatability.

When features are defined in terms of local neighborhood operations applied to an image, a procedure commonly referred to as feature extraction, one can distinguish between feature detection approaches that produce local decisions whether there is a feature of a given type at a given image point or not, and those who produce non-binary data as result. The distinction becomes relevant when the resulting detected features are relatively sparse. Although local decisions are made, the output from a feature detection step does not need to be a binary image. The result is often represented in terms of sets of (connected or unconnected) coordinates of the image points where features have been detected, sometimes with subpixel accuracy.

When feature extraction is done without local decision making, the result is often referred to as a feature image. Consequently, a feature image can be seen as an image in the sense that it is a function of the same spatial (or temporal) variables as the original image, but where the pixel values hold information about image features instead of intensity or color. This means that a feature image can be processed in a similar way as an ordinary image generated by an image sensor. Feature images are also often computed as integrated step in algorithms for feature detection.

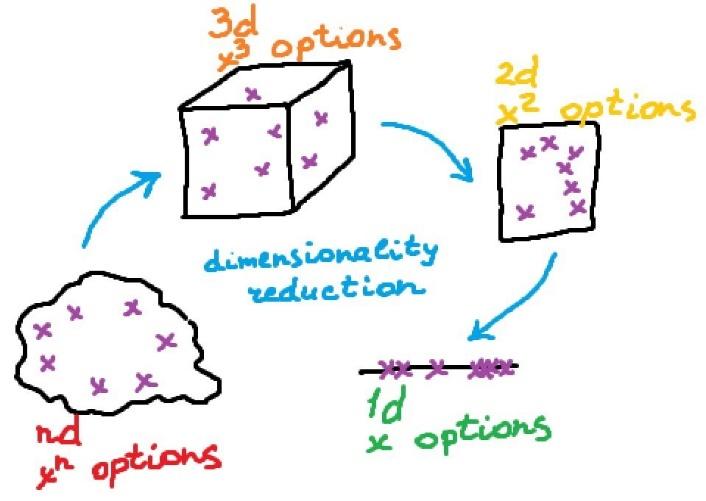
### **Feature vectors and feature spaces**

In some applications, it is not sufficient to extract only one type of feature to obtain the relevant information from the image data. Instead two or more different features are extracted, resulting in two or more feature descriptors at each image point. A common practice is to organize the information provided by all these descriptors as the elements of one single vector, commonly referred to as a feature vector. The set of all possible feature vectors constitutes a feature space.

A common example of feature vectors appears when each image point is to be classified as belonging to a specific class. Assuming that each image point has a corresponding feature vector based on a suitable set of features, meaning that each class is well separated in the corresponding feature space, the classification of each image point can be done using standard [classification](https://en.wikipedia.org/wiki/Statistical_classification) method.

Another and related example occurs when [neural network](https://en.wikipedia.org/wiki/Artificial_neural_network)-based processing is applied to images. The input data fed to the neural network is often given in terms of a feature vector from each image point, where the vector is constructed from several different features extracted from the image data. During a learning phase, the network can itself find which combinations of different features are useful for solving the problem at hand.

**Feature Reduction:**



**Dimensionality reduction**, or **dimension reduction**,

is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its [intrinsic dimension](https://en.wikipedia.org/wiki/Intrinsic_dimension). Working in high-dimensional spaces can be undesirable for many reasons; raw data are often [sparse](https://en.wikipedia.org/wiki/Sparse_matrix) as a consequence of the [curse of dimensionality](https://en.wikipedia.org/wiki/Curse_of_dimensionality), and analyzing the data is usually [computationally intractable](https://en.wikipedia.org/wiki/Computational_complexity_theory#Intractability). Dimensionality reduction is common in fields that deal with large numbers of observations and/or large numbers of variables, such as [signal processing](https://en.wikipedia.org/wiki/Signal_processing), [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition), [neuroinformatics](https://en.wikipedia.org/wiki/Neuroinformatics), and [bioinformatics](https://en.wikipedia.org/wiki/Bioinformatics).

Methods are commonly divided into linear and nonlinear approaches. Approaches can also be divided into [feature selection](https://en.wikipedia.org/wiki/Feature_selection) and [feature extraction](https://en.wikipedia.org/wiki/Feature_extraction). Dimensionality reduction can be used for [noise reduction](https://en.wikipedia.org/wiki/Noise_reduction), [data visualization](https://en.wikipedia.org/wiki/Data_visualization), [cluster analysis](https://en.wikipedia.org/wiki/Cluster_analysis), or as an intermediate step to facilitate other analyses.

## Feature selection

[Feature selection](https://en.wikipedia.org/wiki/Feature_selection) approaches try to find a subset of the input variables (also called features or attributes). The three strategies are: the filter strategy (e.g. [information gain](https://en.wikipedia.org/wiki/Information_gain_in_decision_trees)), the wrapper strategy (e.g. search guided by accuracy), and the embedded strategy (selected features are added or removed while building the model based on prediction errors).

[Data analysis](https://en.wikipedia.org/wiki/Data_analysis) such as [regression](https://en.wikipedia.org/wiki/Regression_analysis) or [classification](https://en.wikipedia.org/wiki/Statistical_classification) can be done in the reduced space more accurately than in the original space.

**Feature projection** (also called feature extraction) transforms the data from the [high-dimensional space](https://en.wikipedia.org/wiki/High-dimensional_space) to a space of fewer dimensions. The data transformation may be linear, as in [principal component analysis](https://en.wikipedia.org/wiki/Principal_component_analysis) (PCA), but many [nonlinear dimensionality reduction](https://en.wikipedia.org/wiki/Nonlinear_dimensionality_reduction) techniques also exist.[[4]](https://en.wikipedia.org/wiki/Dimensionality_reduction#cite_note-4)[[5]](https://en.wikipedia.org/wiki/Dimensionality_reduction#cite_note-5) For multidimensional data, [tensor representation](https://en.wikipedia.org/wiki/Tensor_representation) can be used in dimensionality reduction through [multilinear subspace learning](https://en.wikipedia.org/wiki/Multilinear_subspace_learning)

## What is Feature Reduction?

Feature reduction, also known as dimensionality reduction, is the process of reducing the number of features in a resource heavy computation without losing important information. Reducing the number of features means the number of variables is reduced making the computer’s work easier and faster. Feature reduction can be divided into two processes: [feature selection](https://deepai.org/machine-learning-glossary-and-terms/feature-selection) and [feature extraction](https://deepai.org/machine-learning-glossary-and-terms/feature-extraction). There are many techniques by which feature reduction is accomplished. Some of the most popular are generalized discriminant analysis, [autoencoders](https://deepai.org/machine-learning-glossary-and-terms/autoencoder), non-negative matrix factorization, and [**Principal Component Analysis**](https://deepai.org/machine-learning-glossary-and-terms/principal-components-analysis) **(PCA)**.

**Applications of feature reduction:**

The purpose of using feature reduction is to reduce the number of features (or variables) that the computer must process to perform its function. Feature reduction leads to the need for fewer resources to complete computations or tasks. Less computation time and less storage capacity needed means the computer can do more work. During [machine learning](https://deepai.org/machine-learning-glossary-and-terms/machine-learning), feature reduction removes multicollinearity resulting in improvement of the machine learning model in use.

Another benefit of feature reduction is that it makes data easier to visualize for humans, particularly when the data is reduced to two or three dimensions which can be easily displayed graphically. An interesting problem that feature reduction can help with is called the [curse of dimensionality](https://deepai.org/machine-learning-glossary-and-terms/curse-of-dimensionality). This refers to a group of phenomena in which a problem will have so many dimensions that the data becomes sparse. Feature reduction is used to decrease the number of dimensions, making the data less sparse and more statistically significant for machine learning applications.

Imp Ref:

<https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/>

<https://en.wikipedia.org/wiki/Dimensionality_reduction>

<https://neptune.ai/blog/dimensionality-reduction>

# **Dimensionality Reduction — Does PCA really improve classification outcome?**

# **imp reference:**

<https://towardsdatascience.com/dimensionality-reduction-does-pca-really-improve-classification-outcome-6e9ba21f0a32>

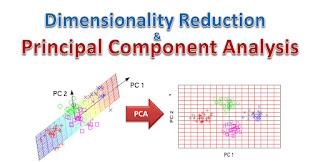
<https://www.kdnuggets.com/2020/05/dimensionality-reduction-principal-component-analysis.html>

<https://towardsdatascience.com/principal-component-analysis-for-dimensionality-reduction-115a3d157bad>

# **Dimensionality Reduction and Principal Component Analysis (PCA)**

We would explain the concept of dimensionality reduction in a very simple way. In line with the principle of my articles, I would try to be as clear as possible.  
In this lesson, we would focus on the explaining the concept and in the next lesson, we would look at the underlying derivation of the technique.  
 Content

* Problem with High Dimensional Data
* What is Dimensionality Reduction
* Two Type of Dimensionality Reduction
* What is Principal Components Analysis
* Methods of Dimensionality Reduction
* How PCA Works
  + Demonstration of PCA
  + Obtain Covariance Matrix
  + Obtain Eigen Pairs
  + Obtain Scores and Loadings
* Extract the Principal Components
* Summary



The Simple Explanation  
You already know that if you are given data in two dimension, say x and y, you could probably plot the graph and see the relationship. What if you are given data in three dimension? You could still try to create the plot but if the data is large enough, visualizing the plot would be difficult.  
Now what if the data is in 10 or 20 or even 100  and more? How could you plot it? Even if you could, you find out that it may not make much sense. This is were dimensionality reduction or dimensional reduction comes in.

Formal Definition of Dimensionality Reduction  
“…is the process of reducing the number of random variables under consideration by obtaining a set of principal variables” – Wikipedia  
“… it the process of reducing the number of variables or features in review” – Big Data University

Problem of High-Dimensional Data

* training a model with high-dimensional data requires much time-space complexity
* Overfitting
* Not all the features of the data are relevant to the problem being solved
* Data in lower dimension has lower noise(unnecessary parts of the data)

Type of Dimensionality Reduction  
The two types of dimensionality reduction are:  
1. Feature Extraction: This technique has to do with finding new features in the data after it has been transformed from a high-dimensional space to a low dimensional space.

2. Feature Selection: This have to do with finding the most relevant features to a problem. This is done by obtaining a subset or key features of the original variables

Methods of Dimesionality Reduction

Principal Component Analysis(PCA): This is a classical method that provides a sequence of best linear approximations to a given high-dimensional observation. It is one of the most popular dimensionality reduction techniques. However, its effectiveness is limited by its global linearity/

Multidimensional Scaling(MDS): This technique is closely related to PCA and have the same limitations as PCA.

Factor Analysis: This technique assumes that the underlying manifold is a linear subspace.

Independent Component Analysis(ICA): This technique starts from a factor analysis solution and searches for rotations that lead to independent components.

Principal Component Analysis (PCA)

PCA is a variance-maximising technique that projects the original data  onto a direction that maximizes variance. PCA performs a linear mapping of the original data to a lower-dimensional space such that the variance of the data in the low-dimensional representation is maximized.

How PCA Works

In math terms, PCA is the performed by carrying out and eigen-decomposition of the co-variance matrix.

The result would be a set of eigenvectors and a set of eigenvalues which can then be used to describe the original data.

A Little More Details  
An eigenvector in linear algebra is a vector that would not change its direction under associated linear transformation. If we  have a non-zero vector v, then its an eigenvector of a square matrix A is Av is a scalar multiple of v.

The eigenvalue is a scalar characteristic value associated with the eigenvector v  
Eigenvectors are the coefficients attached to the eigenvectors and that is what gives the axes their magnitude.

Summary  
Dimensionality Reduction reduces data in high dimension to lower dimension by obtaining the principal components

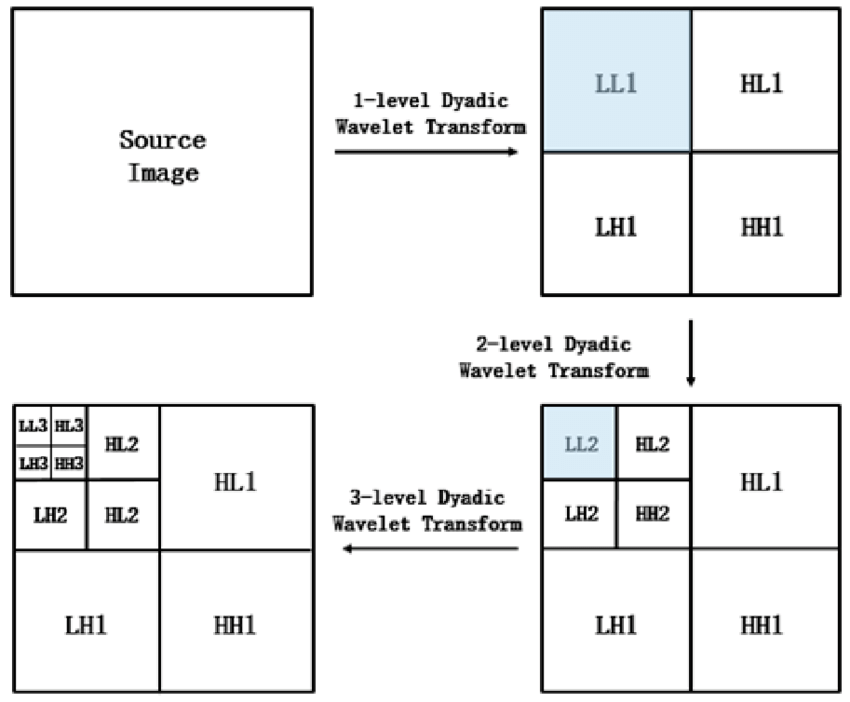
PCA is performed by:

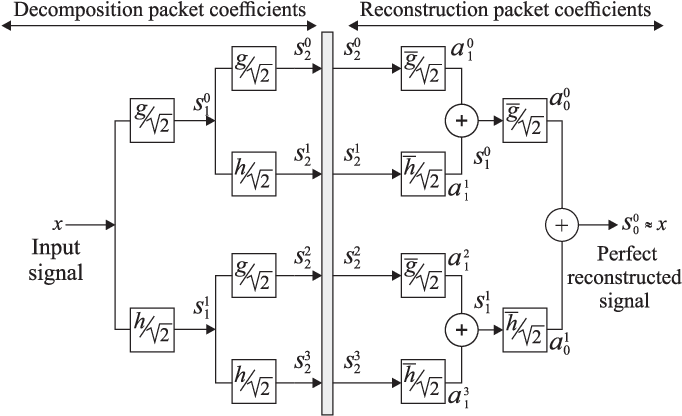
* constructing a co-variance matrix
* performing an eigen-decomposition of that matrix to obtain a set of eigenvectors (W)
* columns of W are ordered by the size of their corresponding eigenvalues
* choose the first n columns of W and use it to describe your data

In the next lesson(which would be a web video), we would actually some of the derivations behind Principal Component Analysis).

# **Discrete wavelet transform**

For Dig ref—2nd Unit PPT., DWT decomposition





Link for refence:

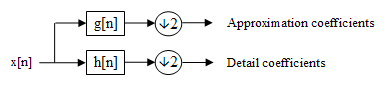
https://en.wikipedia.org/wiki/Discrete\_wavelet\_transform

In [numerical analysis](https://en.wikipedia.org/wiki/Numerical_analysis) and [functional analysis](https://en.wikipedia.org/wiki/Functional_analysis), a discrete wavelet transform (DWT) is any [wavelet transform](https://en.wikipedia.org/wiki/Wavelet_transform) for which the [wavelets](https://en.wikipedia.org/wiki/Wavelet) are discretely sampled. As with other wavelet transforms, a key advantage it has over [Fourier transforms](https://en.wikipedia.org/wiki/Fourier_transform) is temporal resolution: it captures both frequency and location information (location in time).

# **One level of the transform**

The DWT of a signal is calculated by passing it through a series of filters. First the samples are passed through a [low pass filter](https://en.wikipedia.org/wiki/Low_pass_filter) with [impulse response](https://en.wikipedia.org/wiki/Impulse_response) resulting in a [convolution](https://en.wikipedia.org/wiki/Convolution) of the two:

The signal is also decomposed simultaneously using a [high-pass filter](https://en.wikipedia.org/wiki/High-pass_filter) . The outputs give the detail coefficients (from the high-pass filter) and approximation coefficients (from the low-pass). It is important that the two filters are related to each other and they are known as a [quadrature mirror filter](https://en.wikipedia.org/wiki/Quadrature_mirror_filter).



Block diagram of filter analysis

However, since half the frequencies of the signal have now been removed, half the samples can be discarded according to Nyquist’s rule. The filter output of the low-pass filter in the diagram above is then [subsampled](https://en.wikipedia.org/wiki/Downsampling) by 2 and further processed by passing it again through a new low- pass filter and a high- pass filter with half the cut-off frequency of the previous one,i.e.:

This decomposition has halved the time resolution since only half of each filter output characterises the signal. However, each output has half the frequency band of the input, so the frequency resolution has been doubled.

With the [subsampling operator](https://en.wikipedia.org/wiki/Downsampling)

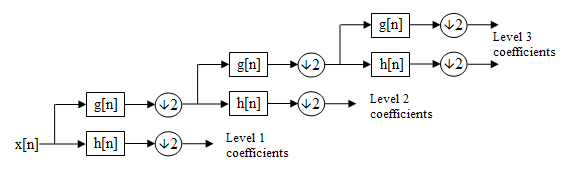
the above summation can be written more concisely.

However computing a complete convolution with subsequent downsampling would waste computation time.

The [Lifting scheme](https://en.wikipedia.org/wiki/Lifting_scheme) is an optimization where these two computations are interleaved.

### **Cascading and filter banks**

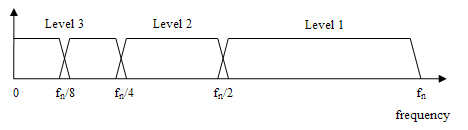
This decomposition is repeated to further increase the frequency resolution and the approximation coefficients decomposed with high and low pass filters and then down-sampled. This is represented as a binary tree with nodes representing a sub-space with a different time-frequency localisation. The tree is known as a [filter bank](https://en.wikipedia.org/wiki/Filter_bank).



A 3 level filter bank

At each level in the above diagram the signal is decomposed into low and high frequencies. Due to the decomposition process the input signal must be a multiple of where is the number of levels.

For example a signal with 32 samples, frequency range 0 to and 3 levels of decomposition, 4 output scales are produced:



**Important:**

**Example After image decomposition to 2nd level – the LL component will have all low frequency component, which can be used in several applications.**

**These low frequency components could be used as 1- feature**

https://towardsdatascience.com/principal-component-analysis-part-1-the-different-formulations-6508f63a5553

CNN is a neural network that extracts **input image features** and another neural network classifies the image features. The input image is used by the feature extraction network. The extracted feature signals are utilized by the neural network for classification.

<https://www.analyticsvidhya.com/blog/2017/06/transfer-learning-the-art-of-fine-tuning-a-pre-trained-model/>

**CNN - ResNet -50 for feature extraction**

Why is ResNet so good?

Using ResNet has **significantly enhanced the performance of neural networks with more layers** and here is the plot of error% when comparing it with neural networks with plain layers. Clearly, the difference is huge in the networks with 34 layers where ResNet-34 has much lower error% as compared to plain-34

<https://www.sciencedirect.com/topics/computer-science/feature-extraction-network>

<https://www.sciencedirect.com/science/article/pii/S0888327017306064>

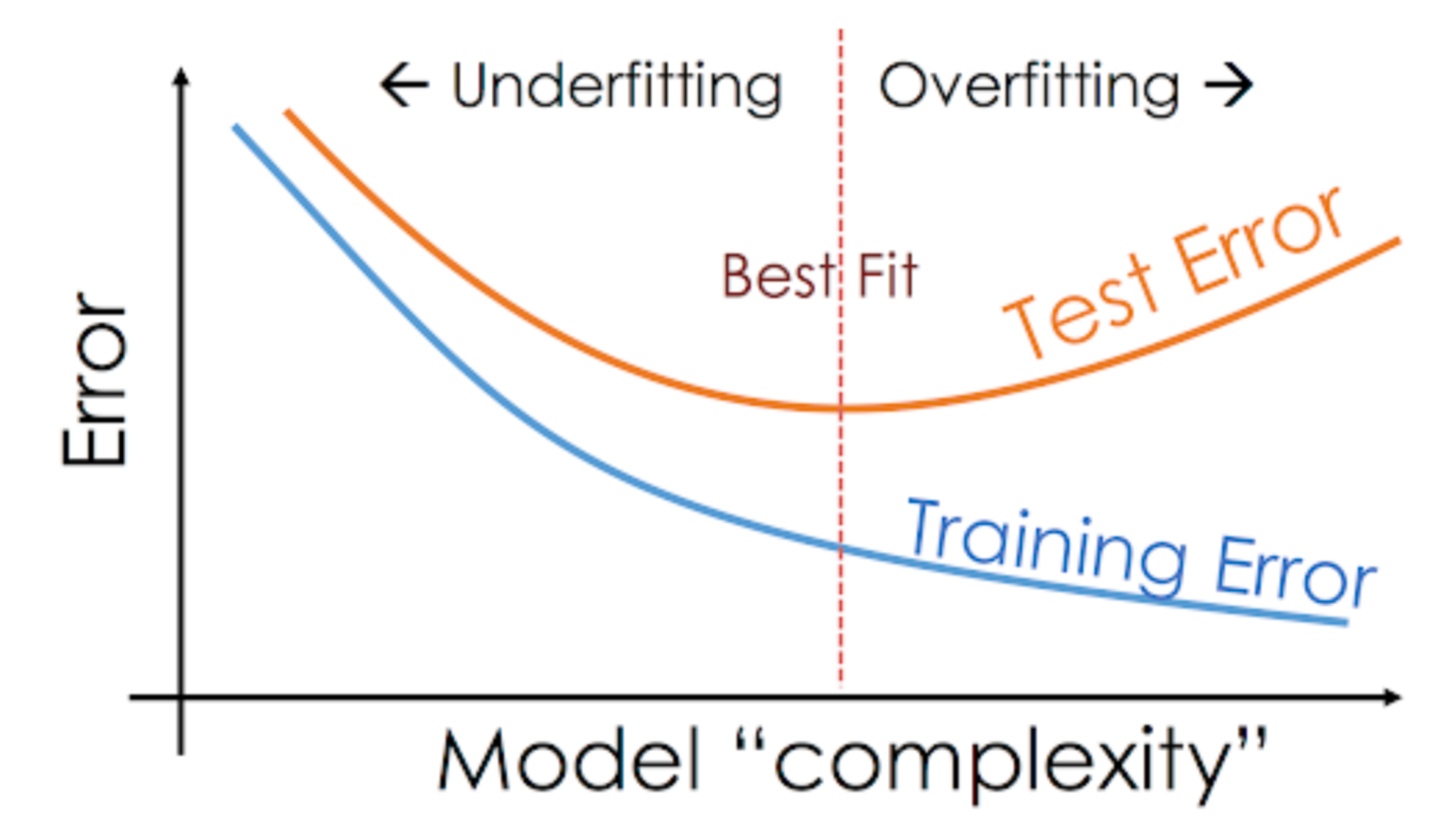
# **Training and Test Sets: Splitting Data**

Link Imp:

<https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7>

https://developers.google.com/machine-learning/crash-course/training-and-test-sets/splitting-data

# **Underfitting vs. Overfitting (vs. Best Fitting) in Machine Learning**



**Link Imp:**

https://www.analyticsvidhya.com/blog/2020/02/underfitting-overfitting-best-fitting-machine-learning/

**Pixel and object based digital image classification?**

<https://www.spiedigitallibrary.org/journals/journal-of-applied-remote-sensing/volume-7/issue-01/073512/Comparison-between-pixel--and-object-based-image-classification-of/10.1117/1.JRS.7.073512.full?SSO=1>

<https://www.google.com/search?q=example+digital+image+Pixel+and+object-based+classification+in+biomedical+image&client=firefox-b-d&sxsrf=APq-WBtdSLkN-mtinMaVP57IASPAy9k0BA%3A1644907961569&ei=uU0LYpOtIrjhz7sP3uG-wAk&ved=0ahUKEwjT7fLVj4H2AhW48HMBHd6wD5gQ4dUDCA0&uact=5&oq=example+digital+image+Pixel+and+object-based+classification+in+biomedical+image&gs_lcp=Cgdnd3Mtd2l6EAM6BwgAEEcQsAM6BwgjEK4CECdKBAhBGABKBAhGGABQoQRYpTBgxjJoAXABeACAAc8BiAH5GpIBBjAuMTUuNZgBAKABAcgBCMABAQ&sclient=gws-wiz>

**What is pixel based image classification?**

In pixel-based classification, **individual image pixels are analysed by the spectral information that they contain** (Richards, 1993). ... The three schemes all use some notion of “distance” to the mean of the class to decide which class to assign pixels.

**What is object-based image classification?**

Introduction. In contrast to pixel-based classification methods that classify individual pixels directly, object-based classification first **aggregates image pixels into spectrally homogenous image objects using an image segmentation algorithm** and then classifies the individual objects.

**CNN pre trained model applications**:

1. Feature extraction
2. Segmentation ( **U-net**)
3. Classification

A pre-trained model is **a model that was trained on a large benchmark dataset to solve a problem similar to the one that we want to solve**. Accordingly, due to the computational cost of training such models, it is common practice to import and use models from published literature (e.g. VGG, Inception, MobileNet).

**Very Deep Convolutional Networks for Large-Scale Image Recognition(VGG-16)** **The VGG-16** is one of the most popular pre-trained models for image classification. Introduced in the famous ILSVRC 2014 Conference, it was and remains THE model to beat even today.

# **Which CNN architecture is best?**

# **https://www.pluralsight.com/guides/introduction-to-resnet**

# **Top 10 CNN architectures**

# **AlexNet. In 2012, Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton won the ImageNet Large Scale Visual Recognition Challenge with a test accuracy of 84.6%³. ...**

# **VGG-16. ...**

# **VGG-19. ...**

# **Inception and GoogLeNet. ...**

# **ResNet. ...**

# **Squeeze Net. ...**

# **DenseNet. ...**

# **Shuffile Net.**

# **Ground-glass opacification**

Link :

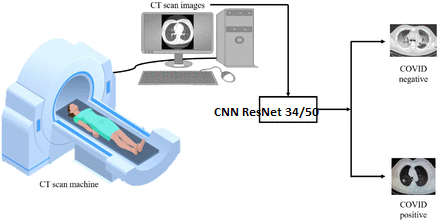
https://radiopaedia.org/articles/ground-glass-opacification-3

**Ground-glass opacity (GGO)** is a **finding seen on chest x-ray** (radiograph) or computed tomography (CT) imaging of the lungs. It is typically defined as an area of hazy opacification (x-ray) or increased attenuation (CT) due to air displacement by fluid, airway collapse, fibrosis, or a neoplastic process.

**What is ground-glass opacity in COVID-19?**

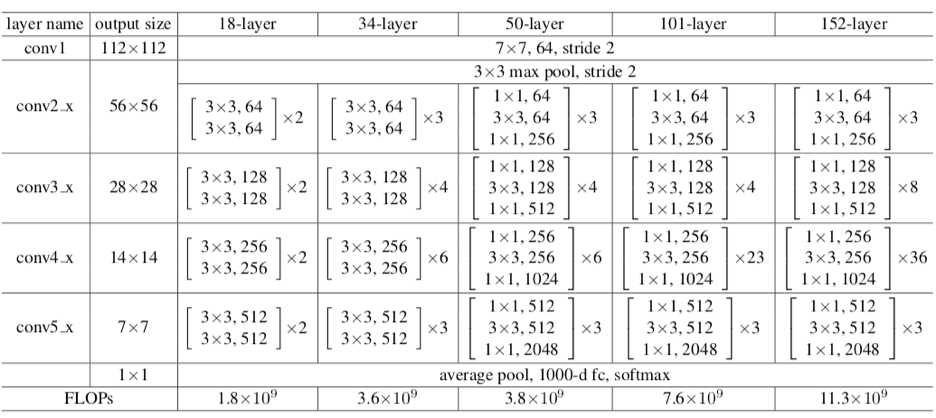
Objectives: Ground-glass opacity (GGO) - **a hazy, gray appearing density on computed tomography** (CT) of lungs - is one of the hallmark features of SARS-CoV-2 in COVID-19 patients. This AI-driven study is focused on segmentation, morphology, and distribution patterns of GGOs

Explain the feature extraction using pre trained CNN models? Infer how pre trained CNN model extract features automatically.



**Write Little details about ResNet 34/50**

[**https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8**](https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8)

****

**Link for reading:**

<https://www.nature.com/articles/s41598-021-93832-2>

<https://www.medrxiv.org/content/10.1101/2020.07.11.20151332v1.full>

**Write Details on Classification parameters:**

Accuracy, Precision and Recall

**Link for reading:**

<https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>