# Transfer Learning

Transfer learning is a machine learning technique where a pre-trained model on one job is utilized as a starting point for training a model on a separate but related task. The goal is to take advantage of the information gained from a large dataset and apply it to enhance a model's performance on a new task or smaller dataset. The fundamental idea behind transfer learning is that the properties that a model picks up on one job may be applicable to another, particularly if the tasks are connected. The pre-trained model serves as a feature extractor, feeding a new model that has been trained on a different task with the output of an intermediate layer. The pre-trained model's weights can be fixed during training, or they can be fine-tuned to better fit the new data.

# Types of Transfer Learning

There are various forms of transfer learning, including:

1. The pre-trained model is utilized as a fixed feature extractor in this sort of transfer learning, and the output of some intermediate layer is used as input to a new model that is trained on a different task. The weights of the pre-trained model are not modified throughout training.
2. The pre-trained model is utilized as a starting point in this sort of transfer learning, and its weights are changed during training to better fit the new data. The pre-trained model's weights are used to initialize the new model, which is then trained on the new job.

Our approach will be more inclined towards the type 2 in this study

# Benefits of Transfer Learning

Transfer learning has several benefits, including:

lowering the data and computational requirements for training: Transfer learning enables us to apply the knowledge of the pre-trained model rather than beginning from scratch. This is particularly helpful when working with small datasets or when the cost of acquiring more data is high.

Enhancing a model's performance: The new model can learn more quickly and effectively by drawing on the prior information, especially when the new task is connected to the prior work.

Several applications, including speech and picture recognition, natural language processing, and even gaming, have effectively used transfer learning. It is a potent machine learning tool that can help enhance model performance while sparing time and resources during training.

# Transfer Learning and NLP

Due to the hierarchical nature of language, transfer learning is especially helpful in natural language processing (NLP), where lower-level features (like words) can be merged to create higher-level features (e.g., phrases, sentences, and documents). Pre-trained models can learn representations from this structure that can be applied to a variety of NLP applications. Strong language models like GPT and BERT, which can be customized for various downstream tasks like text categorization, named entity recognition, and question answering, have been created in NLP via transfer learning. These models are typically pre-trained using unsupervised learning approaches like language modelling or masked language modelling on massive volumes of text data, such as Wikipedia or news articles. Once pre-trained, the models can be fine-tuned on specific downstream tasks by adding a task-specific layer and training it with labeled data.

# Transfer Learning and Computer Vision

Transfer learning has also been extensively employed in computer vision to boost models' performance on a variety of tasks, including object identification, picture classification, and semantic segmentation. Large picture datasets like ImageNet have been used to train pre-trained models like VGG, Inception, and ResNet, and their weights have been made available to the public. These pre-trained models can be refined for particular tasks via transfer learning or utilized as feature extractors. A pre-trained model like VGG, for instance, can be utilized as a feature extractor in image classification by swapping out its final layer with a new one and retraining it on a new dataset with a different set of classes. In object detection, a pre-trained model such as Faster R-CNN can be used as a feature extractor, and its last few layers can be fine-tuned for object detection on a new dataset.

# Transfer Learning and Computer Vision in Medical Field

Transfer learning has revolutionized computer vision by enabling the development of precise models even with sparse data. As a result, breakthroughs in areas like autonomous driving, facial recognition, and medical imaging have been made possible. Transfer learning has a number of significant advantages in medical imaging. First off, the scale of medical imaging datasets is frequently constrained, and producing big, labelled datasets can be challenging and time-consuming. Without the requirement for a significant amount of labelled data, transfer learning enables medical personnel to use pre-trained models that have already learnt to extract pertinent features from images. Second, transfer learning enables health care experts to create models that may be used for a variety of medical imaging tasks. A model that has been taught to recognize tumors in one kind of medical image, for instance, may be adjusted to recognize tumors in another kind of medical image. This can significantly cut down on the time and resources needed to create new models for various jobs. Furthermore, transfer learning can enhance the precision and dependability of medical picture analysis. Medical professionals can create models that are more accurate and robust by utilizing pre-trained models, which lowers the possibility of mistakes and misdiagnosis.

# VGG19

A deep neural network architecture called VGG19 was presented in 2014 by researchers at the University of Oxford. It is used to classify images. It is a variation of the VGG network that the same researchers first developed, but with a more complex architecture. 16 convolutional layers, 3 fully connected layers, and one final SoftMax output layer make up the 19 layers of the VGG19 architecture. The network has a stride of 1 pixel for all convolutions and a modest filter size of 3x3 pixels for each convolutional layer, allowing it to maintain the input image's spatial resolution. The spatial dimensions of the feature maps are decreased by the interleaving of max-pooling layers with the convolutional layers.

The depth of the VGG19 architecture, which enables it to extract more abstract and complicated properties from input images, is one of its important characteristics. The ImageNet dataset, which contains over 1 million photos and 1000 item categories, was used to train the network. The network's weights were first initialized with random values during training, and they were then improved using the backpropagation technique and the cross-entropy loss function. For transfer learning in a range of computer vision tasks, such as image classification, object recognition, and image segmentation, the VGG19 architecture has been extensively employed as a pre-trained network. VGG19 has emerged as a well-liked benchmark for assessing the performance of new deep learning models due to its robust performance on the ImageNet dataset and deep architecture.

# Experiment

In this study, we have used VGG19 (a transfer learning – computer vision model) to classify kidney stones. There are two classes, one with stone and the other is without stone. The data that has been used is opensource and can be accessed at: <https://github.com/yildirimozal/Kidney_stone_detection>. We have finetuned VGG19 by

* Unfreezing the last convolution block
* Chopping off the dense classifier.

When using transfer learning, i.e., adapting a pre-trained model to a new task, it is often necessary to remove the dense layers. The reason for this is that the dense layers of a pre-trained VGG model are typically designed for the specific classification task it was originally trained on, which may not be the same as the new task you want to use the model for. By removing the dense layers, you effectively cut off the part of the model that produces the final classification output, and you can add your own custom dense layers that are tailored to your specific task. This allows you to fine-tune the pre-trained convolutional layers for your task while keeping the learned feature representations intact.

When using transfer learning with a pre-trained VGG model, after removing the dense layers, you need to fine-tune the model for your specific task. Fine-tuning typically involves updating the weights of some or all of the convolutional layers in the model using a new dataset. In general, it is a good practice to freeze the weights of the early convolutional layers of the pre-trained model because these layers typically learn low-level features that are relevant to a wide range of image recognition tasks. By keeping these early layers frozen, you can leverage the pre-trained model's ability to extract useful feature representations from the images without overfitting to the new dataset. On the other hand, the later convolutional layers of the pre-trained model tend to learn more high-level features that are more specific to the original classification task that the model was trained on. By unfreezing these higher layers, you allow the model to learn task-specific features that are relevant to your new dataset.

In this code implementation, we have taken some of the code from the original TensorFlow documentation and their tutorial for finetuning transfer learning models available at: <https://www.tensorflow.org/tutorials/images/transfer_learning>