**METHODOLOGY OF SDP**

**THEORETICAL STUDIES**

**DEEP LEARNING**

Deep learning is a subset of [machine learning](https://www.ibm.com/cloud/learn/machine-learning), which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to “learn” from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

There are different types of neural networks to address specific problems or datasets.

* [***Convolutional neural networks (CNNs)****,*](https://www.ibm.com/cloud/learn/convolutional-neural-networks)used primarily in computer vision and image classification applications, can detect features and patterns within an image, enabling tasks, like object detection or recognition. In 2015, a CNN bested a human in an object recognition challenge for the first time.
* [***Recurrent neural network (RNNs)***](https://www.ibm.com/cloud/learn/recurrent-neural-networks)are typically used in natural language and speech recognition applications as it leverages sequential or times series data.

**PRE TRAINEDMODEL RESNET 50**

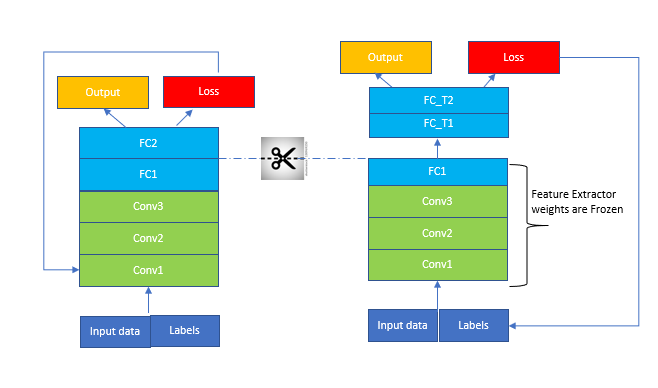
A pre-trained model has been previously trained on a dataset and contains the weights and biases that represent the features of whichever dataset it was trained on. Learned features are often transferable to different data. For example, a model trained on a large dataset of bird images will contain learned features like edges or horizontal lines that you would be transferable your dataset.

Pre-trained models are beneficial to us for many reasons. By using a pre-trained model you are saving time. Someone else has already spent the time and compute resources to learn a lot of features and your model will likely benefit from it.

**Transfer learning using Pre-trained model as Feature Extractor**

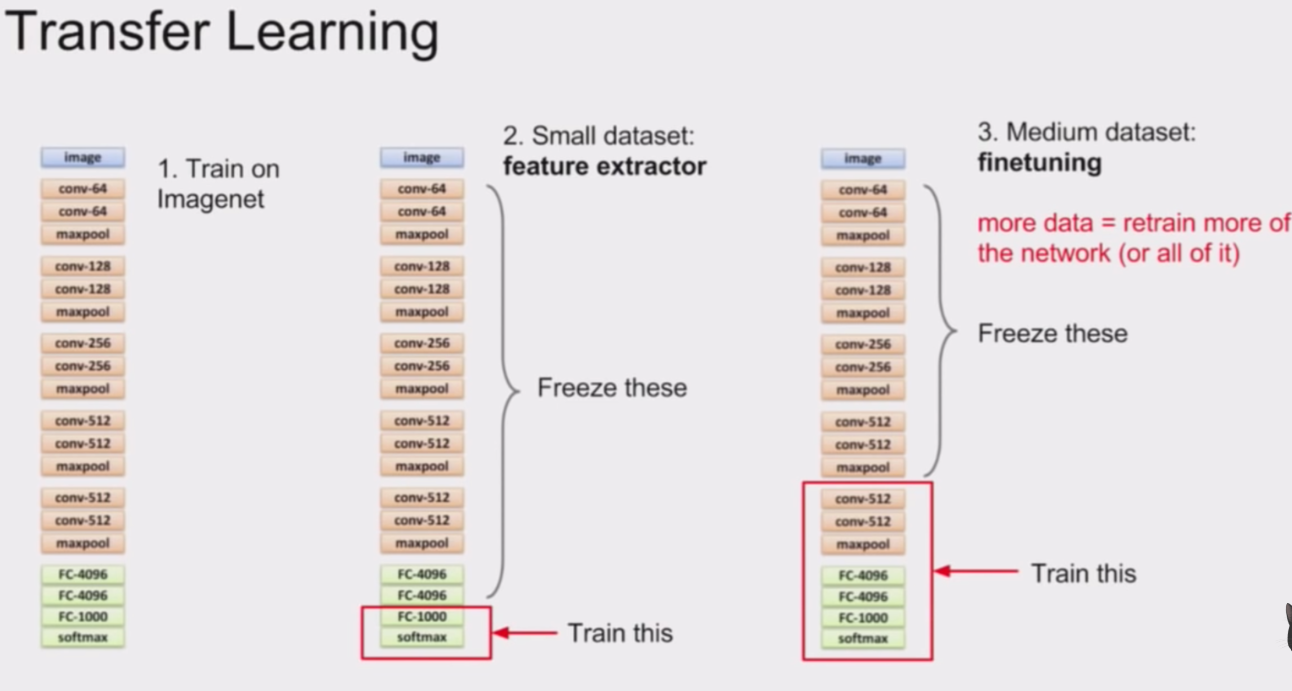
We use ResNet50 deep learning model as the pre-trained model for feature extraction for Transfer Learning.

* To implement Transfer learning, we will remove the last predicting layer of the pre-trained ResNet50 model and replace them with our own predicting layers. FC-T1 and FC\_T2 as shown below
* Weights of ResNet50 pre-trained model is used as feature extractor
* Weights of the pre-trained model are frozen and are not updated during the training.

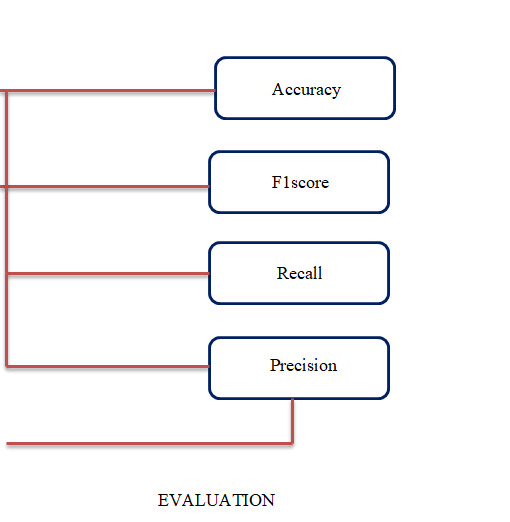


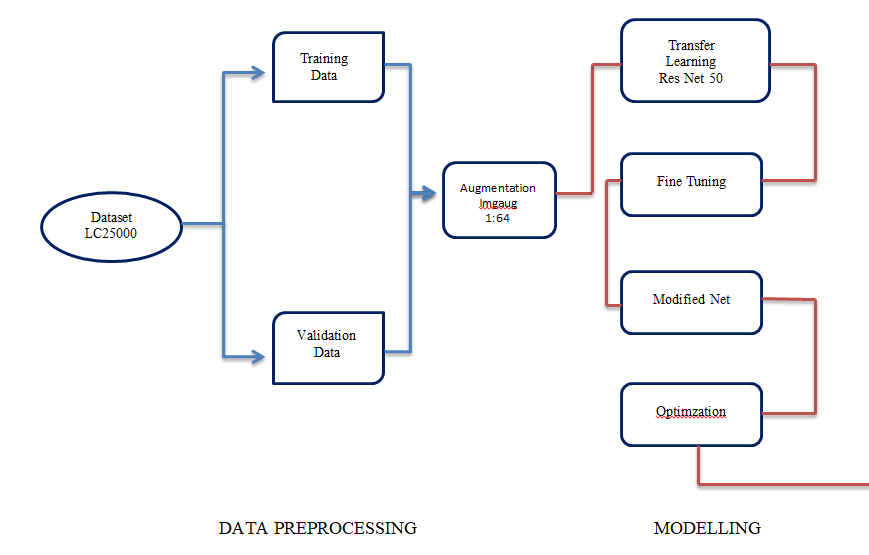
**TRANSFER LEARNING**

Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. For example, knowledge gained while learning to recognize cars could apply when trying to recognize trucks.

****

**EXPERIMENTAL SETUP**





**OVERVIEW**

1. **COLAB**

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

* Zero configuration required
* Free access to GPUs
* Easy sharing

With Colab we can import an image dataset, train an image classifier on it, and evaluate the model, all in just [a few lines of code](https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/quickstart/beginner.ipynb). Colab notebooks execute code on Google's cloud servers, meaning you can leverage the power of Google hardware, including [GPUs and TPUs](https://colab.research.google.com/#using-accelerated-hardware), regardless of the power of your machine. All you need is a browser.

Colab is used extensively in the machine learning community with applications including:

* Getting started with TensorFlow
* Developing and training neural networks
* Experimenting with TPUs
* Disseminating AI research
* Creating tutorials

1. **Data Processing**

**Data Split**

The dataset used in this proposal is LC25000. The dataset contains five classes, three for lung cancer and two for colon cancer. The scope of this study is limited to the lung cancer detection, so only 15000 images of three classes of lung cancer will be used in this study. The three classes of lung cancer include adenocarcinoma (aca), squamous cell carcinoma (scc), and benign (n). Each class has 5000 images. For the purpose of training, and validation the dataset is being divided into two chunks. The bigger chunks has 80% of the data from all three classes and is called training data. Remaining data (20%) is divided for validation and in all classes.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tasks | Training data 80% | | | Test data 20% | | |
| ACA | SCC | Ben | ACA | SCC | Ben |
| Lung cancer | 2000 | 2000 | 4000 | 500 | 500 | 1000 |
| Subtype | 4000 | 4000 |  | 1000 | 1000 |  |

*Table 2: Data split*

**Data Augmentation**

It is generally accepted that more data trains the model better. If there is less data then model either start overfitting or give unexpected results. In medical imaging we can leverage the advantage of data augmentation techniques without worrying about skewness or introducing error in the dataset. The LC25000 dataset is already augmented for 750 lung images with left and right rotation (upto 25 degrees) and horizontal and vertical flips as well. We can use built-in complex data augmentation library “imgaug” to augment data upto desired images. 1:64 augmentation is applied in our case.

## MODELLING

* **Fine Tuning of pre-trained model**

Pre-trained feature extraction will be applied in this study. The concept of transfer learning is that weight of pre-defined models are trained on generic dataset and we input our data and the weight are adapt corresponding to our input data. There are huge benefits of using pre-trained models and they give exceptional results in wide range of cases, along with requirement of less computational capability.

In our case, we used ResNet50 for feature extraction and the only task at this step is the fine tuning of the pre-trained model i.e. ResNet50. Fine tuning of the model requires setting the hyper-parameters. And, wide research in literature is still unable to figure out a proper way to come up with appropriate values of hyper-parameters. So, we are only left with the hit and trial method to set the values of ResNet50’s hyper-parameters.

* **Modified CNN**

After fine tuning of the model hyper-parameters, we will add customized layers at the end of the transfer-learning. Three layers are added Max-pooling2D, Average-pooling2D and flatten. The output of the flatten layer will give feature vector, which is supplied to the output layers having sigmoid activation. Beside all that a dropout layer with dropout rate of 0.5 is also applied.

* **Optimization**

In the end, optimizer is also required to reduce computational complexity and optimum training of the model. There are wide range of optimizers being used in the domain deep learning for wide range of applications. Some of the most common among them are gradient descent, stochastic gradient descent, adadelta, RMSProp, ADAM, etc. In our case, we will be using ADAM, as it is the standard and gives brilliant output in wide range of deep learning models.

**METHOD OF ANALYSIS**

The model designed and applied in the above heading will be evaluated on four metrics i.e. precision, recall, f1 score, and accuracy.

Relevant percentages of these models will evaluate the performance of the model. In order to better understand these evaluation metrics, we should look into the below formulas.

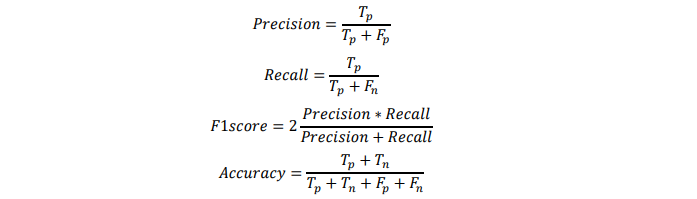
𝑇𝑝 = 𝐶𝑜𝑟𝑟𝑒𝑐𝑡 𝑝𝑜𝑠𝑖𝑡𝑖𝑣𝑒 𝑐𝑙𝑎𝑠𝑠 𝑝𝑟𝑒𝑑𝑖𝑐𝑡𝑖𝑜𝑛

𝑇𝑛 = 𝐶𝑜𝑟𝑟𝑒𝑐𝑡 𝑛𝑒𝑔𝑎𝑡𝑖𝑣𝑒 𝑐𝑙𝑎𝑠𝑠 𝑝𝑟𝑒𝑑𝑖𝑐𝑡𝑖𝑜𝑛

𝐹𝑝 = 𝑊𝑟𝑜𝑛𝑔 𝑝𝑜𝑠𝑖𝑡𝑖𝑣𝑒 𝑐𝑙𝑎𝑠𝑠 𝑝𝑟𝑒𝑑𝑖𝑐𝑡𝑖𝑜𝑛

𝐹𝑛 = 𝑊𝑟𝑜𝑛𝑔 𝑛𝑒𝑔𝑎𝑡𝑖𝑣𝑒 𝑐𝑙𝑎𝑠𝑠 𝑝𝑟𝑒𝑑𝑖𝑐𝑡𝑖𝑜𝑛

From the above definition of the terminology, the metrics of evaluations can be derived as follow:



The model is expected to achieve performance more than 50% on all the above metrics.

**RESULT EXPECTED**

Statistical models based on the extracted features will stratified NSCLC patients into high-risk and low-risk groups and will also the type of lung cancer.