METASTATIC CANCER DETECTION USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

in

Electronics and Telecommunication

by

Chevula Srinivasulu	**(16ET13)**
Khan Akbar Aslam	**(16ET19)**
Shaikh Anwari Jahan	**(17DET35)**
Ansari Fahim Ahemd	**(16ET09)**

Under the guidance of

Prof. Chaya.S

Department of Electronics and Telecommunication Engineering

Anjuman-I-Islam's Kalsekar Technical Campus

Sector 16, New Panvel, Navi Mumbai

University of Mumbai

2019-2020

CERTIFICATE

Department of Electronics and Telecommunication Engineering
Anjuman-I-Islam's Kalsekar Technical Campus
Sector 16, New Panvel , Navi Mumbai
University of Mumbai
2019-2020

This is to certify that the project entitled Metastatic Cancer Detection Using Machine Learning is a bonafide work of **Chevula Srinivasulu (16ET13), Khan Akbar(16ET19), Shaikh Anwari Jahan(17DET35), Ansari Fahim Ahemd(16ET09)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering in Department of Electronics and Telecommunication Engineering.

Supervisor

Examiner

Head of Department

Director

Project Report Approval for Bachelor of Engineering

This project entitled **" Metastatic Cancer Detection Using Machine Learning "** by **Chevula Srinivasulu, Khan Akbar, Shaikh Anwari Jahan, Ansari Fahim Ahmed** is approved for the degree of **Bachelor of Engineering Electronics and Telecommunications**.

Examiner

....

Supervisor

……………………….

Date:

Declaration

We declare that this written submission represents my ideas in my own words and where others ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Chevula Srinivasulu
(16ET13)

……………………

Khan Akbar Aslam
(16ET19)

……………………

Shaikh Anwari Jahan
(17DET35)

……………………..

Ansari Fahim Ahmed
(16ET09)

………………………

Date

Acknowledgments

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

We highly indebted to Prof. Chaya Ravindra for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express my gratitude towards my parents & Staff of Anjuman-I-Islam's Kalsekar Technical Campus for their kind co-operation and encouragement which help me in completion of this project.

We My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

Chevula Srinivasulu	**(16ET13)**
Khan Akbar Aslam	**(16ET19)**
Shaikh Anwari Jahan	**(17DET35)**
Ansari Fahim Ahemd	**(16ET09)**

Abstract

We propose a model for digital pathology segmentation, based on the observation that histopathology images are inherently symmetric under rotation and reflection. Utilizing recent findings on rotation equivariant CNNs, the proposed model leverages these symmetries in a principled manner. We present a visual analysis showing improved stability on predictions, and demonstrate that exploiting rotation equivariance significantly improves tumor detection performance on a challenging lymph node metastases dataset. We further present a novel derived dataset to enable principled comparison of machine learning models. We want to achieve 97% accuracy by using CCN algorithm in machine learning.

Contents

Project Approval for Bachelor of Engineering ………………………………… iii

Declaration ……………………………………………………………………… iv

List of Figures . . . . …. …………………….. ix

Keywords and Glossary……. . . . . ………………... ix

1 Introduction 1

1.1 Statement of Project . . . . ……………………. 4

1.1.2 Motivation .. . . ………… 4

1.2 Objective and Scope . . . . ………………… 5

2 Literature Review 6

2.1 Paper Title . . . . ………………… 6

2.2 Limitation . . . . ……………… 7

2.2.1 How to Overcome…………………………… …………………… 7

3 Technical Details 8

3.1 Methodology . . . . ……………… 6

3.2 Project Requirements . . . . ……………….. 7

3.2.1project code. 13

3.2.3 Hardware & Software Requirements . . . . ……….. . 20

3.2.2 VGG16 Archirecture. 24

3.3 Project Architecture . . . . …………………… 25

4 Market Potential 26

4.1 Market Potential of Project . . . . …………………….. 26

4.2 Competitive Advantages of Project . . . . …………… 26

5 Conclusion and Future Scope 29

5.1 Conclusion . . . . …………….. 26

5.2 Future Scope . . . . ………………… 29

References ………………………………………………………………… 30

List of Figures

1.2 Tumor Divided Cell …………………………………………………………………….. 1

1.2 Cancer Spreads Body Part …………………………………………………………….. 2

1.2 IBM Snap Machine Learning. ……………………………………………………….. 3

3.1 Method. ………………………………………………………………………………… 9

3.1 Unsupervised Learning……………………………………………………………….. 10

3.1 Supervised Learning. . . . .… .. 11

3.1 Processing.. . . . . ……………. 12

3.2.1 VGG16 image convergion………………………………………………………….. 20

3.4 Flow Chart ………………………………………………………………… 25

3.5 Accuracy Score 29

Keywords And Glossary

Keywords

Histopathology, Deep Learning , Machine Learning, Whole Slide Images,

Microscopic Images, Microscopic Images Analysis.

Glossary

- Cancer: a serious disease caused by cells that are not normal and that can spread to one are many parts of the body.
- Histopathology: a branch of pathology concern with the tissues changes characteristic of the disease.
- Metastatic: the separate of the disease producing Agencies (such as cancer cell)
- Deep Learning: it is the branch of Machine Learning in which study a Mathematics & Statics.
- Microscopic: Invisible or indistinguishable without the use of a microscope.

Chapter 1

Introduction

The primary purpose that most cancers is so severe is its ability to unfold in the frame. Cancer cells can spread domestically by using getting into close by ordinary tissue. Cancer can also spread regionally, to close by lymph nodes, tissues, or organs. And it can spread to remote elements of the body. When this happens, it is known as metastatic most cancers. For many sorts of cancer, it is also called level IV (four) cancer. The process with the aid of which cancer cells unfold to other parts of the body is called metastasis.

When determined under a microscope and examined in different ways, metastatic cancer cells have capabilities like that of the primary most cancers and now not like the cells inside the location where the cancer is found. This is how doctors can tell that it's miles most cancers that has spread from every other part of the body.

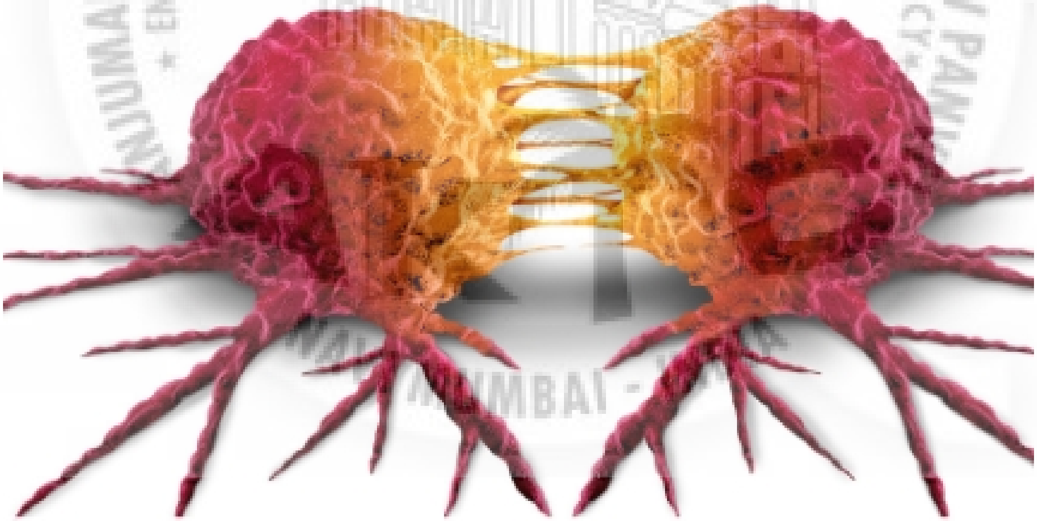


Figure 1 Tumor divided cells

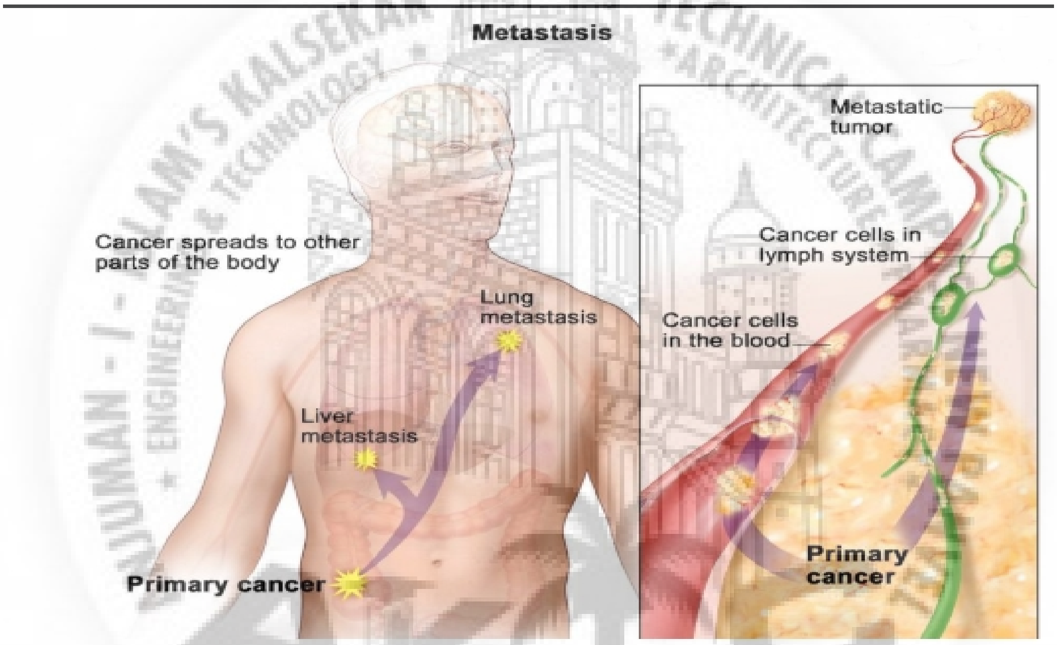

Metastasis
Metastatic tumor
Cancer cells in lymph system
Cancer cells in the blood
Cancer spreads to other parts of the body
Lung metastasis
Liver metastasis
Primary cancer
Primary cancer

Figure 2 Cancer spreads body parts

Metastatic cancer has the equal name because the primary cancer. For instance, breast most cancers that spreads to the lung is known as metastatic breast most cancers, no longer lung cancer. It is handled as stage IV breast most cancers, not as lung cancer.

Cancer cells spread through the body in a series of steps. These steps include:

1. Growing into, or invading, nearby normal tissue.
2. Moving through the walls of nearby lymph nodes or blood vessels.
3. Traveling through the lymphatic system and bloodstream to other parts of the body.

4. Stopping in small blood vessels at a distant location, invading the blood vessel walls, and moving into the surrounding tissue.
5. Growing in this tissue until a tiny tumor forms.
6. Causing new blood vessels to grow, which creates a blood supply that allows the tumor to continue growing.

Common Sites of Metastasis:

Cancer type	**Main Sites of Metastasis**
Bladder	Bone, liver, lung
Breast	Bone, brain, liver, lung
Kidney	Adrenal gland, bone, brain, liver, other lung
Ovary	Liver, lung, peritoneum
Lung	Adrenal gland, bone, brain, liver, other lung
Stomach	Liver, lung, peritoneum
Thyroid	Bone, liver, lung

Machine learning:

Machine learning (ML) is a category of set of rules that lets in software program programs to end up extra correct in predicting effects without being explicitly programmed. The simple premise of machine gaining knowledge of is to build algorithms that could get hold of input facts and use statistical evaluation to predict an output while updating outputs as new information will become available.

Machine gaining knowledge of is a fixed of methods used to create laptop packages that could research from observations and make predictions. Machine mastering makes use of algorithms, regressions, and associated sciences to apprehend statistics. These algorithms can typically be thought of as statistical models and networks.

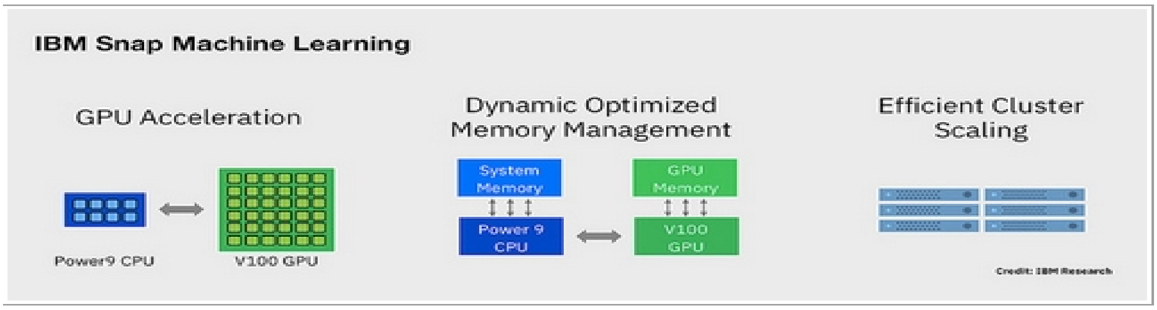


Figure 3 IBM Snap Machine Learning

Deep learning is a part of machine learning. Deep mastering is a selected form of system studying that achieves fantastic electricity and flexibility by means of studying to symbolize the world as nested hierarchy of principles, with every concept defined on the subject of simpler standards, and more summary representations computed in terms of much less summary ones.

Deep getting to know is a subset of system gaining knowledge of strategies. Data is parsed through multiple layers of a deep gaining knowledge of community so that the community can draw conclusions and make choices approximately the statistics. Deep studying strategies permit for remarkable accuracy on large datasets, but those functions make deep getting to know a lot greater resource-intensive than classical gadget gaining knowledge of.

In deep learning we are using **Convolutional Neural Network** (**CNN**, or **ConvNet**). A convolutional neural network (CNN, or ConNet) is a category of deep neural networks, maximum usually implemented to reading visual imagery. They are also referred to as shift invariant or area invariant synthetic neural networks (SIANN), based totally on their shared-weights structure and translation invariance characteristics.

They have packages in image and video reputation, recommender structures, picture type, clinical photo analysis, and herbal language processing.

1.1 Statement of Project

Create an algorithm to identity cancer in small image patches taken from Larger digital pathology scans image. To determine the cancer tumor in given image to develop model using machine learning of CCN model. Develop a Metastatic cancer detection using Machine Learning.

1.1.2 Motivation

Today in our society the most of the people are suffer from cancer disease. The highest increase among all therapy areas, with **cancer** being the most common among Indian people, according to a recent report. As per the **Indian** Council of Medical Research (ICMR) data, **India** had 14 lakh **cancer patients** in 2018 and this number is expected to increase.

1.2 Objective and Future Scope

Simplify the data system by the reducing the number of variables to be studied, without sacrificing too much of accuracy. such a system would help Subsidy Inc. in planning subsidy outlay, monitoring and preventing misuse.

Chapter 2

Literature Review

2.1 Metastatic Cancer Detection Using Supervised Machine Learning

[1] Jason yosinski, jeff clune, yoshua Bengio, and Hod Lipson,"supplementary material for: how transferable are feature in deep neural network". The predictive models discuss here are based on various on supervised ml techniques as well as different input in data sample,combined with the applications of different technique for features selection and classification for provide promising tools for interference in the cancer domain.

[2] Jonthon de matos, Alceu do Souza britto Britto junior, Luiz Eduardo S. Oliveira, Alessandro Lameiras Koerich,''Histopathologic Image Processing: a Review". A study case of breast cancer classification using a mix of deep and shallow machine learning method. The propossed method obtaind accuracy of 91% of the best casse. Our propossed deffer from the others in the point there we intend use a mechanism create better patches the first attempt perform well the little margin of improvement. It was based on the transfer learning one data set of for colorectal histopathology images to other of breast cancer.

[3] Shikha agarwal, jitendra Agarwal,"Neural Network Technique for Cancer Prediction:A Survey" the main aim of this servey in medical diagnosis is to guide reaserche to develop most cost effective user friendly system, process and approches for clinicians. In future study, accuracy of nural network can be enhanced by increasing no. of neurons in the hidden layer. Different training and learning rule can be applied for training ann in order to performance of classsifier.

[4] Jason yosinski, jeff clune, yoshua bengio, od lipson,"How Transferable are feature in deep neural networks". A final surprising result is that inicializing a nework with transferred featurs from always any no. of layers can prodused a boost to genaralization that lingers even after find tunning target data set. It found that initializing with transferd features can improved generalzation a performance even after substitutional find tunning on a new task, which could be a generally usefull technique for improving deep neural network performance.

[5]Bastiaan S.veeling ,jasper Linmans,Jim winkess,"Rotation Equivarint CNNs for igital pathology"Through this dataset,the task of histopthology diagnosis becomes as a challenging benchmark for fundamental machine learning research ,A drive pathch-level dataset is presented ,allowing straightforward and precise evaluation on a challenging histopathology task

[6] M.Tahmooresi ,A.Afshar,B.Bashari Rad ,K.B.Nowshath and M.A.Bamiah,"Early Detection of

breast cancer using machine learning techniques"This study is discusses the data set used for breast cancer detection and diagnosis the proposed model can be used with different data types such as image, blood etc. this method can be applied and tests on another dataset like mammogram and ultrasound to check the performance of difference data type the mammogram was the most frequent data set used compared to other types of data such as ultrasound images, thermal or blood features.

2.2 Limitation

- Used data set have less samples get less accuracy.
- Not using microscope images.
- Developed models are not deploy in websites.

2.2.1 How to overcome

Using large sample data set to get high accuracy.

Using microscope images.

Try to deploy model in websites.

Chapter 3

Technical Details

3.1 Methodology

In our case we analysis the histopathological images using machine learning, The prior information is apply in machine learning algorithm in this technique per-processing should be performed. For example, when cancer regions are detected in WSI, local mini patches around 256 × 256 are sampled from large WSI.

Then feature extraction and classification between cancer and non-cancer are performed in each local patch. The goal of feature extraction is to extract useful information for machine learning tasks.

Various local features such as gray level co-occurrence Matrix (GLCM) and local binary pattern (LBP) have been used for histopathological image analysis, but deep learning algorithms such as convolutional neural network starts the analysis from feature extraction.

Using different software tools to build a model.

1. **TensorFlow** is a free for open source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural network It is used for both research and production at Google.

2. **Pandas** is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

3. **NumPy** is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

4. **Open-CV** (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. The library is cross-platform and free for use. Open CV supports the deep learning frameworks TensorFlow, Torch/PyTorch and Caffe.

5. **Scikit-learn** (formerly scikits. learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Python is an interpreted, high-level, general-purpose programming language. its design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Machine learning technique:

These techniques is a branch of Artificial Intelligence related to study of the sample data set to general idea of mathematical analysis and concept of inference, each AI procedure is comprised of two-stage,

a. Estimation of obscure conditions of the framework in a given informational collection.

b. Utilization of past information to foresee the new predictions.

There are three main type of methods.

a. Unsupervised.

b. Semi- supervised learning

c. Supervised learning.

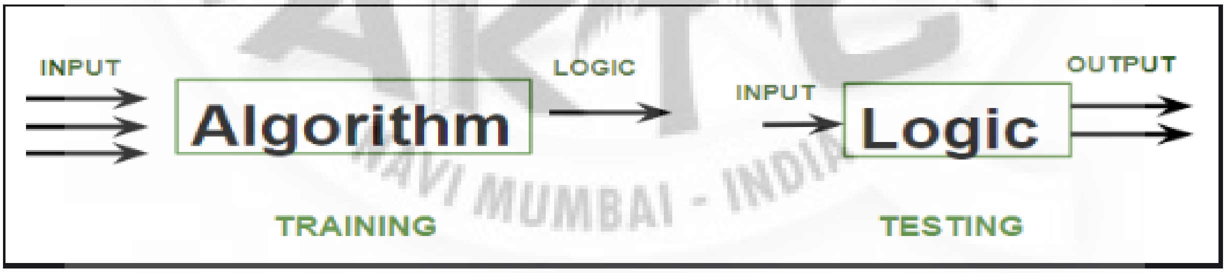


Figure 4 Method

These techniques are utilized in computerized pathology image examination are essentially partitioned into two sorts regulated and unaided AI directed learning is a learning where in the picture is mark related to WSI and label in WSI. The first methods include various calculations like support vector machines, random forest, and convolutional neural systems. The second method is remembered for the concealed structure of the image and without label, k-means, auto encoder, principal component analysis.

a. **Unsupervised:** Unsupervised machine learning algorithms infer patterns from a datasets without reference to known, or labeled, outcomes. Unlike supervised machine learning, unsupervised machine learning methods cannot be directly applied to a regression or a classification problem because you have no idea what the values for the output data might be, making it impossible for you to train the algorithm the way you normally would. Unsupervised learning can instead be used to discover the underlying structure of the data.

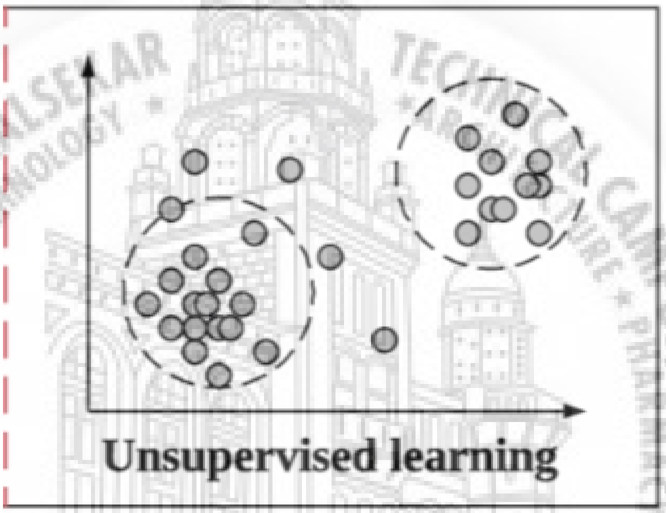


Figure 5 Unsupervised learning

b. **Semi- supervised learning:** Semi-supervised learning is an approach to machine learning that combines a small amount of labeled data with a large amount of unlabeled data during training. Semi-supervised learning falls between unsupervised learning (with no labeled training data) and supervised learning(with only labeled training data).

Unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy. The acquisition of labeled data for a learning problem often requires a skilled human agent (e.g. to transcribe an audio segment) or a physical experiment (e.g. determining the 3D structure of a protein or determining whether there is oil at a particular location). The cost associated with the labeling process thus may render large, fully labeled training sets infeasible, whereas acquisition of unlabeled data is relatively inexpensive. In such situations, semi-supervised learning can be of great practical value. Semi-supervised learning is also of theoretical interest in machine learning and as a model for human learning.

a. **Supervised learning:** Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.[1] It infers a function from labeled training data consisting of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way. The parallel task in human and animal psychology is often referred to as concept learning.

In our project we are using supervised learning.

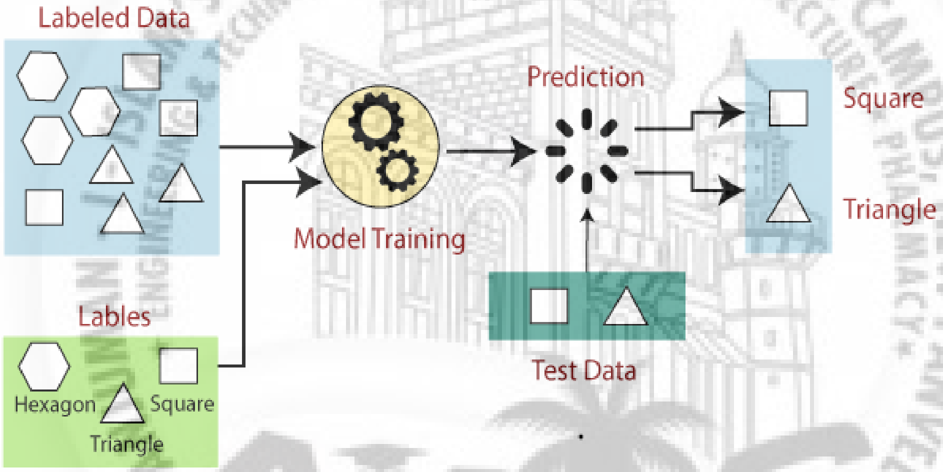


Figure 6 Supervised learning

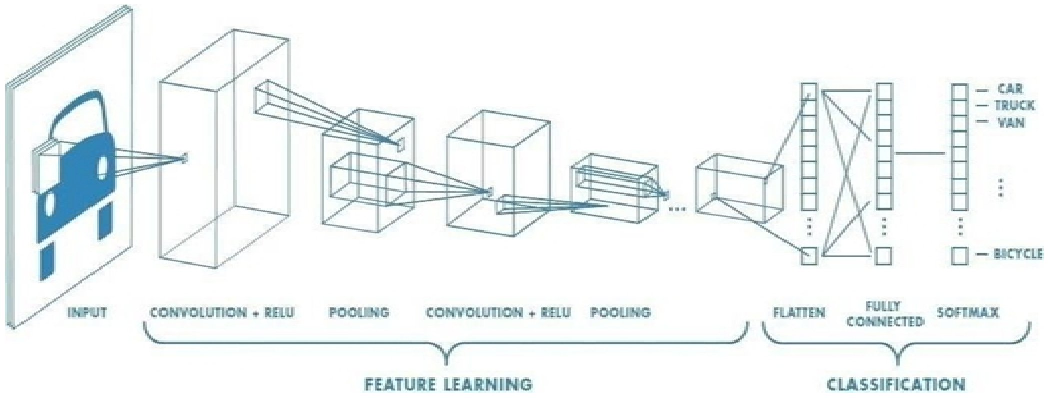


Figure 7 Processing

The CNN modal is classified into two main categories first is feature learning and classification, the feature learning is consisting of different layers, convolution the convolution layer are move the filter or mask to every possible position of the image in each data set, the second layer is a relu layer the function of the relu layer is remove every negative values from the filter images and also it replace with zero, third layer are the pooling layer this layer is compress the large size of images into small size, the final layer is the fully connected layer in this layer the actual classification are happen in this layer take the filter and shrinking images and store in a small list and when we find the new images then proprieties of new images and store images are compare each other and value which is high and they give the result [9,7] .

Our model is designed in a total of three phases :

a. **Stage1**.THE first phase are involves collection of data , the images are collected from kaggle website and some preprocessing are done in phase one.
b. **Stage2.** In preprocessing removal of glare shading are done and identify the texture are dine and feature extraction and segmentation are done. feature extraction like color,shape size etc.
c. **Stage3**.This phase are the most important phase in this phase the building of machine learning model and training and testing of the model for accurate output. [11,9].

3.2 Project Code:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from zip file import ZipFile
file_name=’histopathologic-cancer-detection.zip’
z=ZipFile(file_name)
z.extracall(‘extracted_content’)

import tensorflow as tf
from tensorflow import keras

import os
import shutil import copyfile, move
from tqdm import tqdm
import h5py
import random


from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.model import Sequential
from tensorflow.keras.layers import Dense, Flatten, Dropout, Activation
from tensorflow.keras.layers import BatchNormalization, GlobalAveragepoolind2D
from tensorflow.keras.callback import ModelCheckpoint, ReduceLROnplateau, EarlyStopping
from tensorflow.keras.application import VGG16


dataset_df=pd.read_csv("/content/extracted_content/train_labels.csv")
dataset_df[“file_name”]=[item.id+”.tif” for idx, item in dataset_df.iterrows()]
dataset_df[“groundtruth”] = [“cancerous” if item.label==1 else “healthy” for idx, item in dataset_df.iterrows()]
dataset_df.head()
```

```
training_sample_percentage = 0.8
training_sample_size = int(len(dataset_df)*training_sample_percentage)
validation_sample_size = len(dataset_df)-training_sample_size


training_batch_size = 128
validation_batch_size = 128
target_size = (96,96)


train_datagen = ImageDataGenerator(
rescale=1./255,
horizontal_flip=True,
vertical_flip=True,
zoom_range=0.2,
width_shift_range=0.1,
height_shift_range=0.1
)


train_generator = train_datagen.flow_from_dataframe(
        dataframe = training_df,
        x_col=’filename’,
        y_col=’groundtruth’,
        directory=’/content/extracted_content/train’,
        target_size=target_size,
        batch_size=training_batch_size,
        shuffle=True
        class_mode=’binary’)


validation_datagen = ImageDataGenerator(rescale=1./255)
validation_generator =
        validation.datagen.flow_from_dataframe(dataframe = validationaaa_df,
        x_col=’filename’,
        y_col=’groundtruth’,
        directory=’/content/extracted_content/train’,
        target_size=target_size,
        shuffle=False,
        batch_size=validation_batch_size,
        class_mode=’binary’)


def plot_random_samples(generator):
        generator_size = len(generator)
        index=random.randit(0,generator_size-1)
        image,label = generator.__getitem__(index)

        sample_number = 10
        fig = plt.figure(figsize = (20,sample_number))
        for i in range(0,sample_number):
                ax = fig.add_subplot(2, 5, i+1)
                ax.imshow(image[i])
                if label[i] ==0:
                        ax.set_title(“1”)
                elif label[i] ==1
                        ax.set_title(“0”)
        plt.tight_layout()
        plt.show()



plot_random_sample(validation_generator)



input_shape = (96, 96, 3)
```

```
pretrained_layers = VGG16(weight='imagenet',include_top = False, input_shape=input_shape)
pretrained_layers.summary()


for layer in pretrained_layers.layers[:-8]:
        layer.trainable = False

for layer in pretrained_layers.layers:
        print(layer, layer.trainable)


dropout_dense_layer = 0.6


model = Sequential()
model.add{pretrained_layers}

model.add(GlobalAveragePooling2D90())
model.add(Dense(256, use_bias=False))
madel.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(dropout_dense_layer))


model.add(Dense(1))
model.add(Activation('sigmoid'))


model.summary()
model.load_weights("best_model.h5")
```

Explanation of Different Libraries and modules use in the project

#NumPy: It is a library for the python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical function to operate on these arrays.

#Matplotlib: Matplotlib is a plotting library for the python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into application using general-purpose GUI toolkits like Tkinter, WxPython, Qt, or GTK+.

#Pandas: Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy packages and its key data structure is called the DataFrame. DataFrame allows you to store and manipulate tabular data in rows of observations and columns of variables.

#zip file: zip file work with ZIP archives. The ZIP file format is a common archive and compression standard. This model provides tools to create, read, write, append and list a ZIP file. Any advance use of this module will require and understanding of the format, as defined in PKZIP Application Note.

#TensorFlow: TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning Models directly or by using wrapper libraries that simply the process built on top of TensorFlow.

#Keras: Kera's is TensorFlow's high-level API for building and training deep learning model. it's used for fast prototyping, state-of-the-art research, and production, with three key advantages:

- User-friendly: Keras has a simple, consistent interface optimized for common use cases. It provides clear and actionable feedback for few restrictions.
- Modular and composable: Keras models are made by connecting configurable building block together, with few restrictions.
- Easy to extend: Write custom building block to express new ideas for research. Create new layers, metrics, loss funcions, and develop state-of-the-art models.

#OS : OS Miscellaneous operating system interfaces. This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see open(), if you want to manipulate the paths, see the os.path module, and if you want to read all the lines in all the files on the command line see the fileinput module. For creating temporary files and directories see the tempfile module, and for high-level file and directory handling see the shutil module.

Note on the availability of these functions:

#Shutil: The shutil module offers a number of high-level operations on files and collections of files. In particular, functions are provided which support file copying and removal. For operations on individual files, see also the os module.

#TQDM: tqdm is a progress bar library designed to be fast and extensible. It is written in Python, though ports in other languages are available.

# HDF5: HDF5 (Hierarchical Data Format) is a library for managing the formatting of scientific data. It allows you to store, read, visualize, manipulate and analyze the data in an efficient manner. It supports an unlimited variety of datatypes, and is designed for flexible and efficient I/O and for high volume and complex data. HDF5 is portable and is extensible, allowing applications to evolve in their use of HDF5. The HDF5 Technology suite includes tools and applications for managing, manipulating, viewing, and analyzing data in the HDF5 format. HDF (also known as HDF4) is a library and multi-object file format for storing and managing data between machines. There are two versions of HDF: HDF4 and HDF5. HDF4 is the first HDF format. Although HDF4 is still funded, new users that are not constrained to using HDF4, should use HDF5.

Environment modules:

The following modules providing HDF are available on both Cedar and Graham via CVMFS:

1. **Hdf** : contains HDF of version 4.1 and previous releases.
2. **hdf5** : contains the most recent version of HDF5.
3. **hdf5-mpi** : includes support of MPI.

#ImageDataGenerator Generator batches of tensor image data with real-time data augmentation. The data will be looped over (in batches).

#BatchNormalization in Keras provides support for batch normalization via the BatchNormalization layer. The layer will transform input so that they are standardized, meaning that they will have a mean of zero and a standard deviation of one.

#**Application checkpoint** is a fault tolerance technique for long running processes.
It is an approach where a snapshot of the state of the system is taken in case of system failure. If there is a problem, not all is lost. The checkpoint may be used directly, or used as the starting point for a new run, picking up where it left off.
When training deep learning models, the checkpoint is weight of the model. These weights can be used to make predictions as is, or used as the basis for ongoing training.The Keras library provides a checkpoint capability by a callback API.

#**ReduceLROPlateau** Reduce learning rate when a metric has stopped improving.
Models often benefit from reducing the learning rate by a factor of 2-10 once learning stagnates.
This callback monitors a quality and if no improvement is seen for a 'patience' number of epochs, the learning rate is reduced.

3.2.1 VGG16 architecture: (convolute the images)

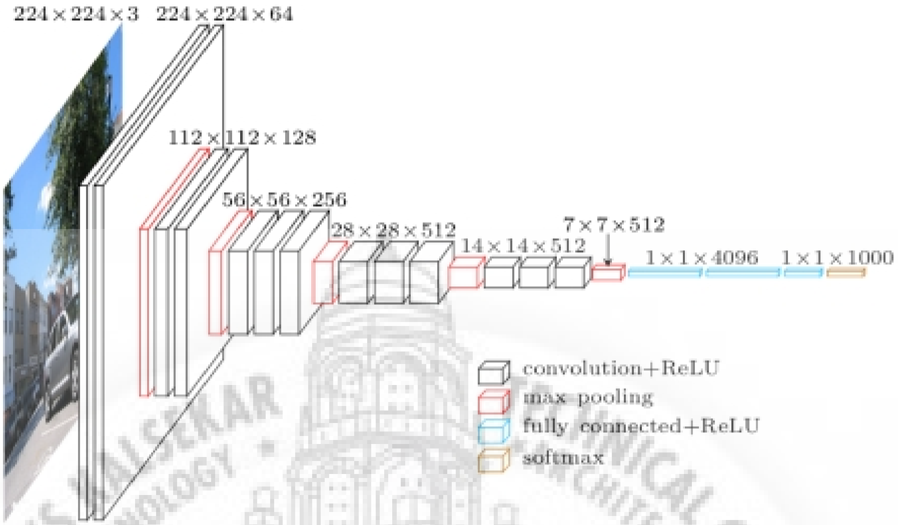


Fig 8: VGG16 image convergion

VGG16 is a convolution neural net (CNN) architecture which was used in ILSVR. It is considered to be one of the excellent vision model architecture. Most unique thing about VGG16 is that instead of having a large number of hyper-parameter they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC(fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights. This network is a pretty large network and it has about 138 million (approx) parameters.

Now to implement full VGG16 from scratch in Keras. This implement will be done on small patches taken from a large scale microscopic images dataset. We are downloaded the data set from Kaggle website.Once you have downloaded the images then you can proceed with the steps written below.

Here I first importing all the libraries which i will need to implement VGG16. I will be using Sequential method as I am creating a sequential model. Sequential model means that all the layers of the model will be arranged in sequence. Here I have imported ImageDataGenerator from keras .preprocessing. The objective of ImageDataGenerator is to import data with labels easily into the model. It is a very useful class as it has many function to rescale, rotate, zoom, flip etc. The most useful thing about this class is that it doesn't affect the data stored on the disk. This class alters the data on the go while passing it to the model.

Here I am creating and object of ImageDataGenerator for both training and testing data and passing the folder which has train data to the object trdata and similarly passing the folder which has test data to the object tsdata. The folder structure of the data will be as follows.

The ImageDataGenerator will automatically label all the data inside cat folder as cat and vis-à-vis for small patches folder. In this way data is easily ready to be passed to the neural network.

Here I have started with initializing the model by specifying that the model is a sequential model. After initializing the model I add
→ 2 x convolution layer of 64 channel of 3x3 kernal and same padding
→ 1 x maxpool layer of 2x2 pool size and stride 2x2
→ 2 x convolution layer of 128 channel of 3x3 kernal and same padding
→ 1 x maxpool layer of 2x2 pool size and stride 2x2
→ 3 x convolution layer of 256 channel of 3x3 kernal and same padding
→ 1 x maxpool layer of 2x2 pool size and stride 2x2
→ 3 x convolution layer of 512 channel of 3x3 kernal and same padding
→ 1 x maxpool layer of 2x2 pool size and stride 2x2
→ 3 x convolution layer of 512 channel of 3x3 kernal and same padding
→ 1 x maxpool layer of 2x2 pool size and stride 2x2
Now add Relu(Rectified Linear Unit) activation to each layers so that all the negative values are not passed to the next layer.

After creating all the convolution I pass the data to the dense layer so for that I flatten the vector which comes out of the convolutions and add
→ 1 x Dense layer of 4096 units
→ 1 x Dense layer of 4096 units
→ 1 x Dense Softmax layer of 2 units

Now use RELU activation for both the dense layer of 4096 units so that I stop forwarding negative values through the network. I use a 2 unit dense layer in the end with softmax activation as I have 2 classes to predict from in the end which are the images is cancer and non-cancer. The softmax layer will output the value between 0 and 1 based on the confidence of the model that which class the images belongs to.

After the creation of softmax layer the model is finally prepared. Now I need to compile the model.

Here Now will be using Adam optimiser to reach to the global minima while training out model. If I am stuck in local minima while training then the adam optimiser will help us to get out of local minima and reach global minima. We will also specify the learning rate of the optimiser, here in this case it is set at 0.001. If our training is bouncing a lot on epochs then we need to decrease the learning rate so that we can reach global minima.

Now can check the summary of the model which I created by using the code.

The output of this will be the summary of the model which I just created.

After the creation of the model I will import Model Checkpoint and EarlyStopping method from keras. Now will create an object of both and pass that as callback functions to fit_generator.

Model-checkpoint helps us to save the model by monitoring a specific parameter of the model. In this case we are monitoring validation accuracy by passing **val_acc** to Model-checkpoint. The model will only be saved to disk if the validation accuracy of the model in current epoch is greater than what it was in the last epoch.

EarlyStopping helps us to stop the training of the model early if there is no increase in the parameter which we have set to monitor in EarlyStopping. In this case I am monitoring validation accuracy by passing **val_acc** to EarlyStopping. I have here set **patience** to 20 which means that the model will stop to train if it doesn't see any rise in validation accuracy in 20 epochs.

Now using model.fit_generator as I am using ImageDataGenerator to pass data to the model. I will pass train and test data to fit_generator. In fit_generator steps_per_epoch will set the batch size to pass training data to the model and validation_steps will do the same for test data. You can tweak it based on your system specifications.

After executing the above line the model will start to train and you will start to see the training/validation accuracy and loss.

Once you have trained the model you can visualise training/validation accuracy and loss. As you may have noticed We passing the output of mode. Fit generator to hist variable. All the training/validation accuracy and loss are stored in hist and I will visualise it from there.

To do predictions on the trained model I need to load the best saved model and pre-process the image and pass the image to the model for output.

Here we have loaded the image using image method in keras and converted it to NumPy array and added an extra dimension to the image to image for matching NHWC (Number, Height, Width, Channel) format of keras.

This is a complete implementation of VGG16 in keras using ImageDataGenerator. We can make this model work for any number of classes by changing the unit of last softmax dense layer to whatever number we want based on the classes which we need to classify.

3.3 Project Requirements

3.3.1 Software Requirements

Anaconda cloud, python libraries (Pandas, Numpy, Pyplotlib, Sklearn, Tensorflow open cv ,spyder , CNN)

Hardware Requirements

- Quad core Intel Core i7 Skylake or higher (Dual core is not the best for this kind of work, but manageable)
- 16GB of RAM (8GB is okay but not for the performance you may want and or expect)
- M.2 PCIe or regular PCIe SSD with at least 256GB of storage, though 512GB is best for performance. The faster you can load and save your applications, the better the system will perform. (SATA III will get in the way of the system's performance)
- Premium graphics cards, so things with GTX 980 or 980Ms would be the best for a laptop, and 1080s or 1070s would be the best for the desktop setup. (try not to sacrifice too much here. While a 980TI or a 970m may be cheaper, this is also a critical part of the system, and you'll see a performance drop otherwise.)

3.4 Project Architecture

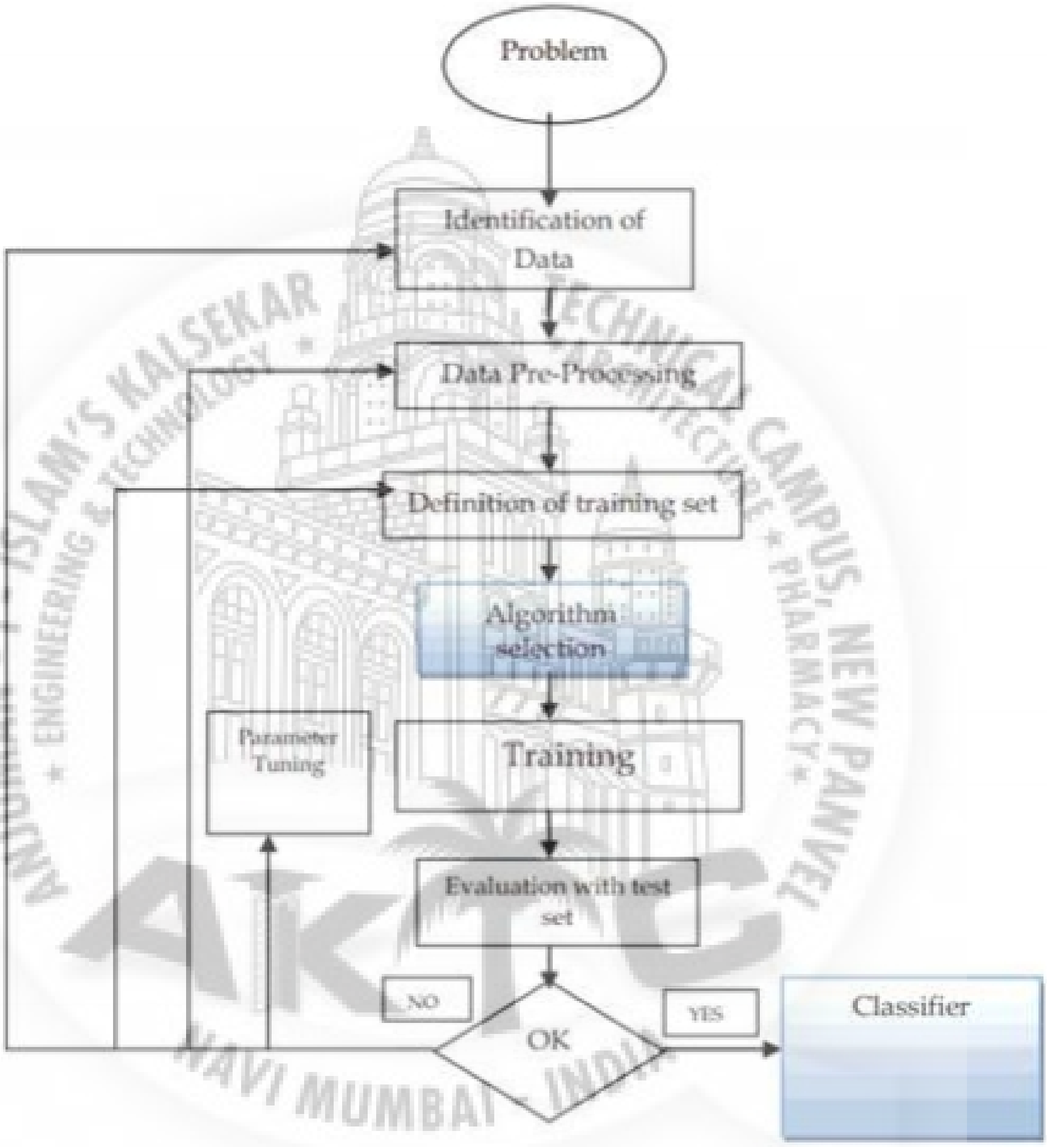


Figure 9 Flow chart

Chapter 4
Market Potential

4.1 Market Potential of Project

In this project the cost of the treatment of cancer patient will be decrease less than 50% and Accuracy will be increased. also it is easily handled and least time are required for treatment of cancer patient and its increased the cancer patient treatment and decrease the cost, time.

It helps the society to protect cancer patient and to save human life from cancer disease Our project also helps the Human being to have a healthy life.

4.2 Competitive Advantages of Project

Due its compact algorithm size and cheaper in cost our project is based on the machine learning and deep learning algorithm also its give the high accuracy and in least cost.

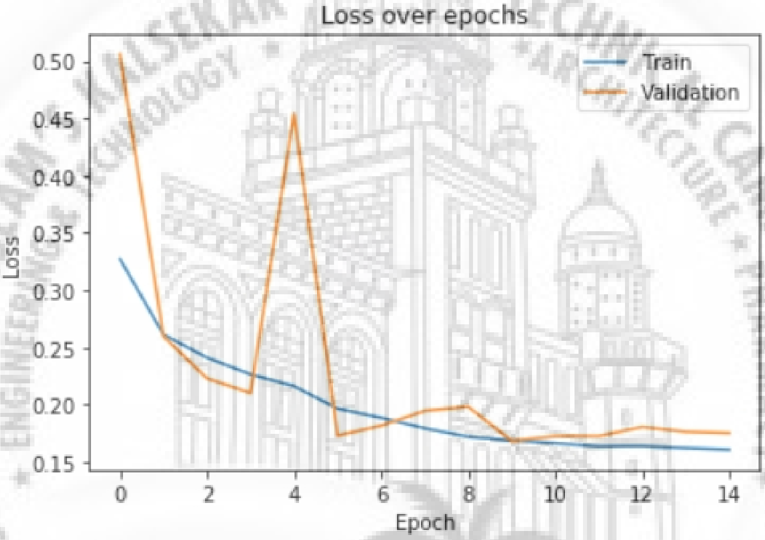
It is also least time processing to give the proper output and after that the treatment. Its required least time to find out where the cancer cell are present in microscopic images.

Chapter 5

Result Conclusion and Future Scope

5.1 Result

In This projects we are using the histopathologic images for cancer detection using deep learning technique known as VGG16 Transfer learning technique but we have trying the different transfer learning technique like VGG16 and Resnet but in this technique we are not able to get proper result, using VGG16 and some image augmentation technique we are able to get the accuracy around 97% which is better than other transfer learning technique



In the given figure when the number of epoch is increase then the training loss and validation loss is decrease at one time the both train and validation loss is constant at that epoch we will get constant accuracy

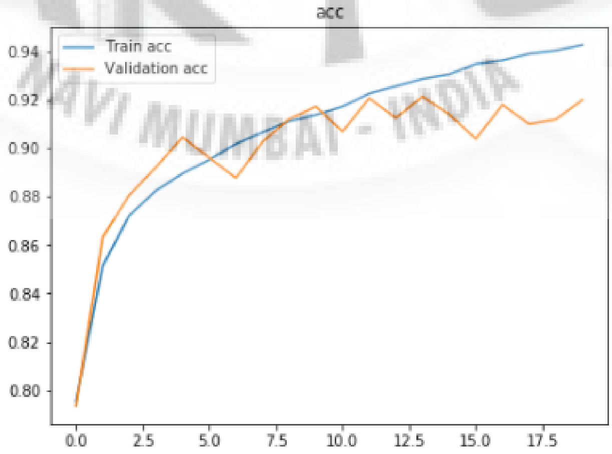


Figure number 10 Accuracy Score

In the given figure we can see that the accuracy is increase Continuous when the number of epoch is increase therefore we can say that our model is performance is good not so over fitting and Under fitting

5.2 Conclusion

We are study different research paper based on machine learning and deep learning using different data sets. We are understand machine learning algorithm work flow and we also see different data sets we are implement pilot projects on machine learning. We are conclude for image processing CCN algorithm suitable for image processing. We develop machine leaning model to detect metastatic cancer with accuracy 97%.

5.3 Future Scope

The future enhancement includes a technology called using Machine Learning Algorithm Metastatic Cancer Detection we are going to take a Microscopic Images to predict any type of cancer. In this technology we are going to implement a website including with the machine learning algorithm we are giving the images and find out the cancer is present or not, also in future we are storing a number of images with the all information related to cancer patient and they find out self-information about that disease how to overcome with proper treatment in a particular time. And that patient will be healthy itself.

In future this technology will be helpful for the common person and they can be done itself treatment using this algorithm and this can be decrease the time to check the patient.

References

[1] John W. Dower Readings compiled for History 21.479. 1991.The Japan Reader Japan 1800-1945 1973: Random House, N.Y.

[2] E. H. Norman Japan’s emergence as a modern state 1940: International Secretariat, Institute of Pacific Relations.

[3] Bob Tadashi Wakabayashi Anti-Foreignism and Western Learning in Early-Modern Japan1986: Harvard University Press.

[4] B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, M. Welling. "Rotation Equivariant CNNs for Digital Pathology". arXiv:1806.03962.

[5] Ehteshami Bejnordi et al. Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. JAMA: The Journal of the American Medical Association, 318(22), 2199–2210. doi:jama.2017.14585.

[6] Kaggle. Histopathologic Cancer Detection — Identify metastatic tissue in histopathologic scans of lymph node sections

[7] Leslie N. Smith. “A disciplined approach to neural network hyper-parameters Part 1 — learning rate, batch size, momentum, and weight decay”.arXiv:1803.09820v2

[8] Shikh Agrawal Jitendra Agrawal neural network techniques for cancer prediction :A survey

[9] Konstantina kourou themis p.exarchos ,Konstantinos p.exarchos Michalis v. karamouzis dimitros Fotiadis Machine learning applications in cancer prognosis and prediction