Project Report

of

‘Skin cancer image recongnition’

for

Data Mining and Analysis (CS-655C)

A Project Report is submitted in partial fulfilment of the requirements for the award of

Bachelor of Engineering

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by
Amandeep, Ashish uadhyay, Deepanshu Garg

(Roll no: CO17310,CO17315,CO17319)

Under the Guidance of

Dr. Varun Gupta (Professor, CSE Dept.)

CHANDIGARH COLLEGE OF ENGINEERING AND TECHNOLOGY

(DEGREE WING)

Government Institute under Chandigarh (UT) Administration, Affiliated to Panjab University, Chandigarh

Sector-26, Chandigarh. PIN-160019

Abstract

In this project the classification of skin cancer images has been presented using CNN. The incidence of skin tumors has steadily increased. Although most are benign and do not affect survival, some of the more malignant skin tumors present a lethal threat if a delay in diagnosis permits them to become advanced. Mostly, an inspection by an expert dermatologist would accurately detect malignant skin tumors in the early stage; however, it is not practical for every single patient to receive intensive screening by dermatologists. To overcome this issue, we have developed a dermatologist-level, computer-aided diagnostics. Recently, the introduction of deep-learning technology, a method that automatically extracts a set of representative features for further classification has dramatically improved classification efficacy which we have used in our project. This technology has the potential to improve the computer classification accuracy of conventional clinical images to the level of skilled dermatologists.

Chapter-1

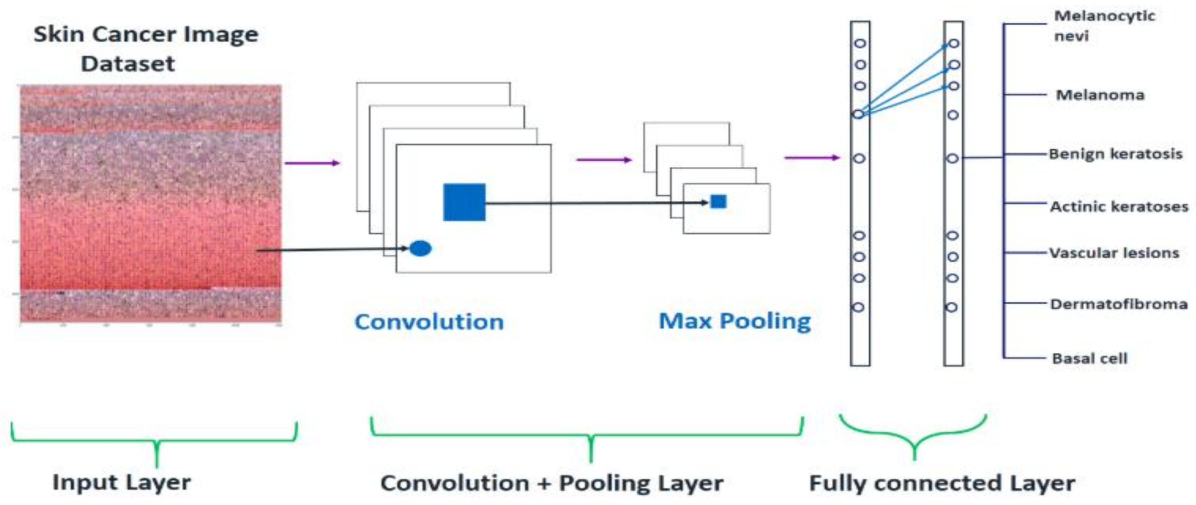
Introduction

Quick, think of five people you love. Your mom, dad, sister or brother, best friend and grandma perhaps? Statistically, 1 of these 5 people will get skin cancer at some point in their life. Keeping true to cancer's deadly reputation, most skin cancers either spread to other parts of the body and/or are fatal unless detected and treated early. And to rain harder on your parade, because of the inefficiency in our healthcare systems today, the diagnosis and treatment will begin in stages much later than it could (and should) be starting.

Skin cancer is the single most common malignancy affecting humankind, with over 5.4 million, in the USA alone, diagnosed on a yearly basis. Yet, diagnosis is still a visual process, which relies on the long-winded procedure of clinical screenings, followed by dermoscopic analysis, and then a biopsy and finally a histopathological examination. This process easily takes months and the need for many medical professionals and still is only ~77% accurate.

The sheer amount of time and technicalities it takes when diagnosing the patient (let alone beginning their treatment) and the many opportunities for human error, leaves thousands dead annually.
But, with the development of artificial intelligence and machine learning capabilities, there is shining potential to spare time and mitigate errors- saving millions of lives in the long run.
In particular, Convolutional Neural Networks (CNNs) can automate most of the diagnosis process with equal or more accuracy than the current methods.

For this project, we replicated a CNN using 10,000 training images to compare the results to human experts and analyze the results to see how they contrast



Chapter-2

Python packages

2.1 Python packages

A package is basically a directory with Python files and a file with the name __init__.py. This means that every directory inside of the Python path, which contains a file named __init__.py, will be treated as a package by Python.

2.2 PACKAGES WE USED

2.2.1 **PANDAS**

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. It is free software released under the three-clause BSD license.[2] The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Library features:

- DataFrame object for data manipulation with integrated indexing.
- Tools for reading and writing data between in-memory data structures and different file formats
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of data sets.
- Label-based slicing, fancy indexing, and subsetting of large data sets.
- Data structure column insertion and deletion.
- Group by engine allowing split-apply-combine operations on data sets.
- Data set merging and joining.
- Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
- Time series-functionality: Date range generation[4] and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging.
- Provides data filtration.

2.2.2 **NumPy**

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.

Features:

NumPy targets the CPython reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays, requiring rewriting some code, mostly inner loops, using NumPy.

2.2.3 **Keras**

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML.Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is François Chollet, a Google engineer. Chollet also is the author of the XCeption deep neural network model

In 2017, Google's TensorFlow team decided to support Keras in TensorFlow's core library.[7] Chollet explained that Keras was conceived to be an interface rather than a standalone machine learning framework. It offers a higher-level, more intuitive set of abstractions that make it easy to develop deep learning models regardless of the computational backend used Microsoft added a CNTK backend to Keras as well, available as of CNTK v2.0

With a variety of supported devices and platforms, you can deploy Keras on any device like

- iOS with CoreML
- Android with Tensorflow Android,
- Web browser with .js support
- Cloud engine
- Raspberry Pi

Fine-Tune Pre-Trained Models in Keras and How to Use Them

Why we use Fine Tune Models and when we use it

Fine-tuning is a task to tweak a pre-trained model such that the parameters would adapt to the new model. When we want to train from scratch on a new model, we need a large amount of data, so the network can find all parameters. But in this case, we will use a pre-trained model so the parameters are already learned and have a weight.

For example, if we want to train our own model to solve a classification problem but we only have a small amount of data, then we can solve this by using a Transfer Learning + Fine-Tuning method.

Using a pre-trained network & weights we don't need to train the whole network. We just need to train the last layer that is used to solve our task as we call it Fine-Tuning method.

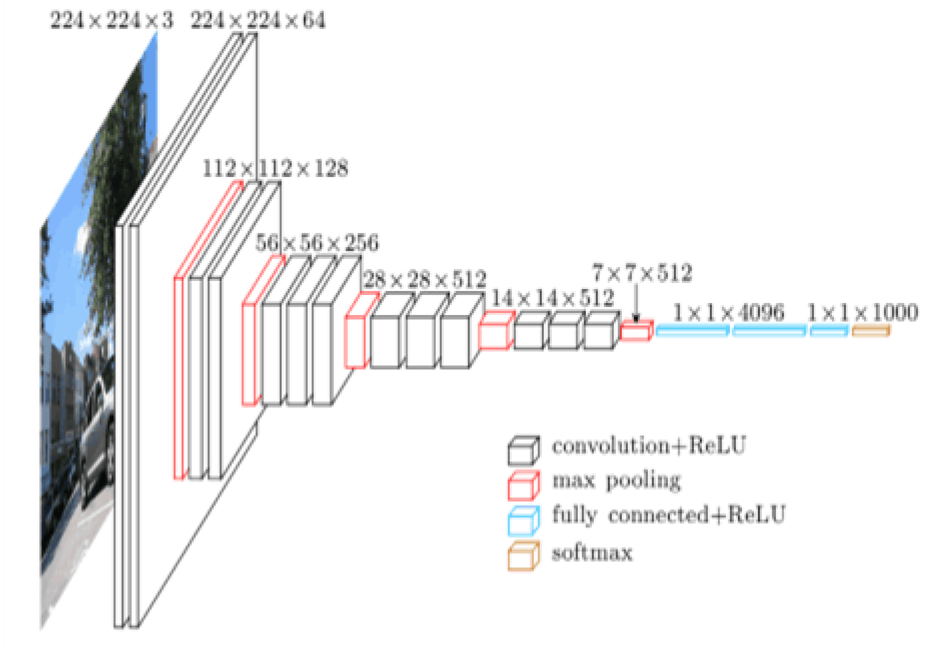
Network Model Preparation

For the pre-trained model, we can load a variety of models that Keras already has in its library such as:

- VGG16
- InceptionV3
- MobileNet
- Xception

- InceptionResNetV2

VGG16 model architecture



2.2.4 **matplot lib**

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 applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged.SciPy makes use of Matplotlib.

Chapter 3

LITERATURE SURVEY

3.1 CONVOLUTIONAL NEURAL NETWORKS

A Convolutional Neural Network (CNN) is composed of 1 or more convolutional layers (often with a subsampling step) so followed by one or more fully connected layers as during a standard multilayer neural network. The architecture of a CNN is meant to require advantage of the 2D structure of an input image. A convolution is that the simple application of a filter to an input that ends up in an activation. Repeated application of the identical filter to an input ends up in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input, like a picture. An advantage of CNNs is that they're easier to coach and have many fewer parameters than fully connected networks with the identical number of hidden units. They need applications in image and video recognition, recommender systems, image classification, medical image analysis, language processing and financial statistics.

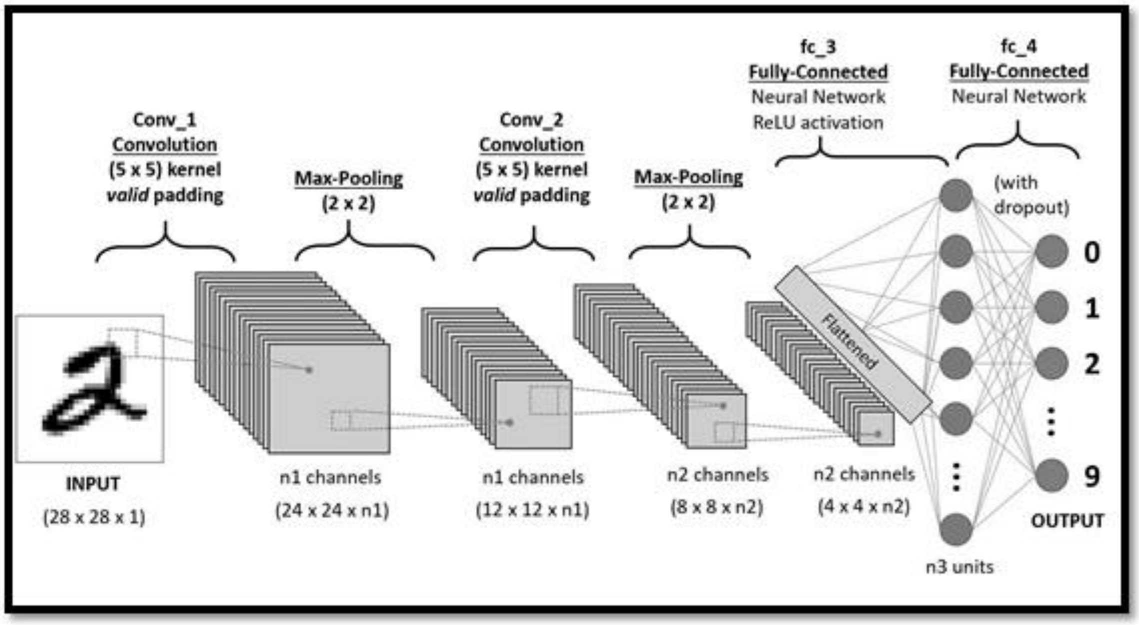


Figure: 3.1

A CNN consists of a variety of convolutional and subsampling layers optionally followed by fully connected layers. The input to a convolutional layer may be a m x m x r image where m is the height and width of the image and r is that the number of channels, e.g. an RGB image has r=3. The convolutional layer will have k filters (or kernels) of size n x n x q where n is smaller than the dimension of the image and q can either be the identical because the number of channels r or smaller and should vary for every kernel. A CNN architecture is made by a stack of distinct layers that transform the input volume into an output volume through a differentiable function. Some distinct styles of layers are commonly used- Convolutional layer, pooling layer, ReLU layer, fully connected layer, Loss layer. The convolutional layer is that the core building block of a CNN. Pooling refers to a kind of non-linear down-sampling. ReLU is the abbreviation of rectified linear measure, which applies the non-saturating activation function. It effectively removes negative values from an activation map by setting them to zero. Neurons in an exceedingly fully connected layer have connections to any or all activations within the previous layer. The "loss layer" specifies how training penalizes the deviation between the anticipated (output) and true labels and is generally the ultimate layer of a neural network.

3.2 IMAGE CLASSIFICATION

Image classification refers to the task of extracting information classes from a multiband raster image. The resulting raster from image classification can be used to create thematic maps. The intent of the classification process is to categorize all pixels in a digital image into one of several land cover classes. The objective of image classification is to identify and portray, as a unique gray level (or color), the features occurring in an image in terms of the object or type of land cover these features actually represent on the ground.

There are three types of image classification techniques- unsupervised classification, supervised classification, object-based image analysis. In unsupervised classification, it first groups pixels into “clusters” based on their properties. Then, each cluster is classified with a land cover class. The two basic steps for unsupervised classification are generate clusters, assign values. In supervised classification, representative samples for each land cover class. The software then uses these “training sites” and applies them to the entire image. The basic steps used are select training areas, generate signature file, classify. Supervised and unsupervised classification is pixel-based. In other words, it creates square pixels and each pixel has a class. But object-based image classification groups pixels into representative vector shapes with size and geometry.

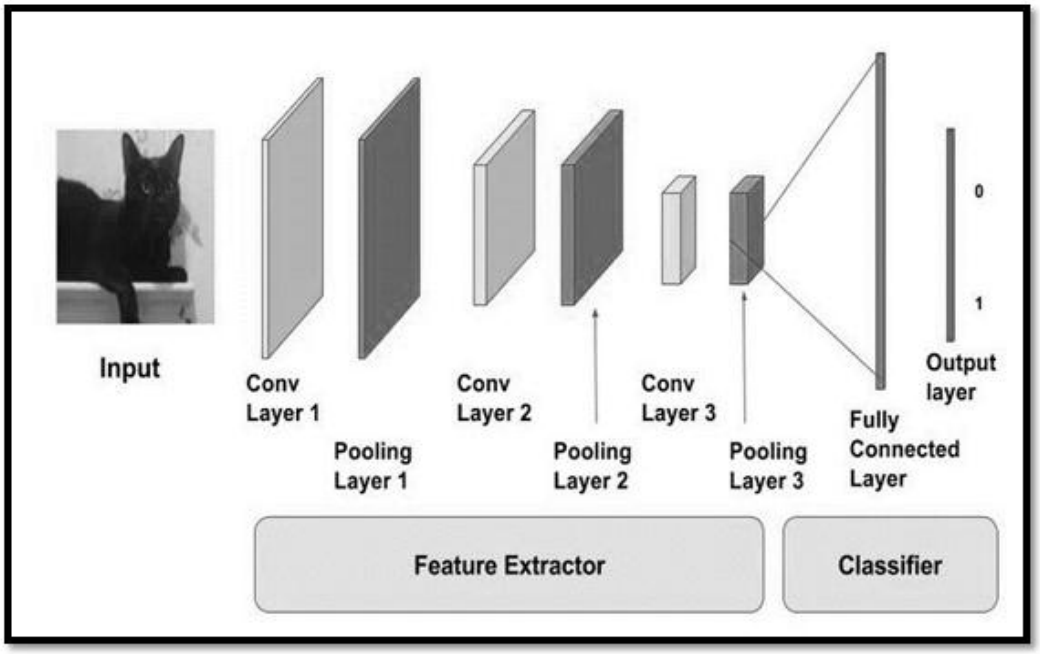


Figure: 2.2

When image is classified using a Convolutional Neural Network, the it is passed through a series of convolutional, nonlinear, pooling layers and fully connected layers, and then generates the output. The Convolution layer is always the first. The image (matrix with pixel values) is entered into it. Next the software selects a smaller matrix there, which is called a filter (or neuron, or core). Then the filter produces convolution, i.e. moves along the input image. The network will consist of several convolutional networks mixed with nonlinear and pooling layers. The nonlinear layer is added after each convolution operation. It has an activation function, which brings nonlinear property. The pooling layer follows the nonlinear layer. It works with width and height of the image and performs a down-sampling operation on them. As a result, the image volume is reduced. After completion of series of convolutional, nonlinear and pooling layers, it is necessary to attach a fully connected layer. This layer takes the output information from convolutional networks.

Chapter 4

Data Source

All data has been collected by a single source which is https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000

The file was downloaded as a single source file and it contains upto about 10k files for training the data in the project.

The whole dataset consists of:

1. 10k files which contains images
2. Images are of different types of skin cancer which occur on different parts of the body.
3. The original size of image was 600*450
4. The size was resized to 100*100 for training and uploading purposes.
5. The image formats is .jpg
6. The images are attached to the file by creating and analyzing a path.

Chapter 5

Working

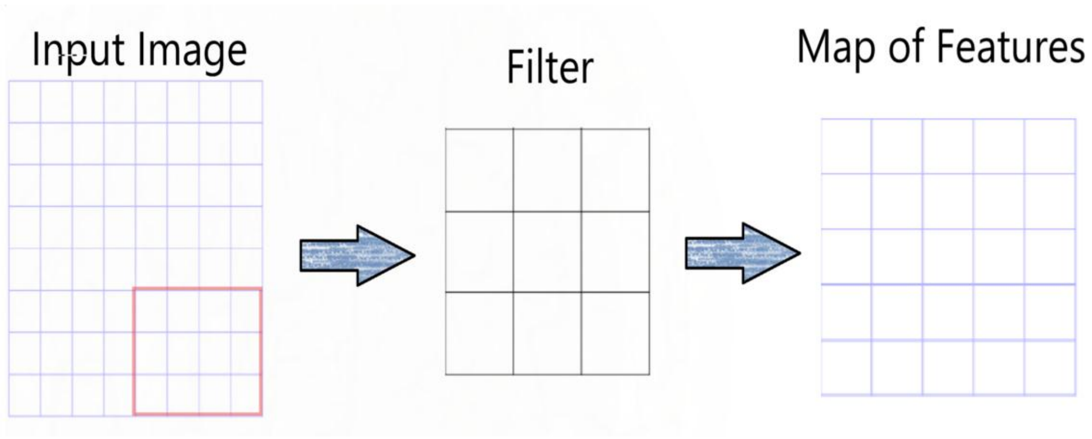
Step 1- Convolution:

A function derived from two given functions by integration which expresses how the shape of one is modified by the other. There are 3 main elements of the convolution operation:

● Input image: the actual image in pixels (basically our input data)

● Feature detector: These can even be called filters. They basically detect certain features in the input image. Feature Detectors are placed over the input image (though are much smaller in size) and count the number of cells in which the feature detector matches a subset of the input image. The feature detector then moves alone the input image to cover all of its areas and the distances it moves can be referred to as “strides”.

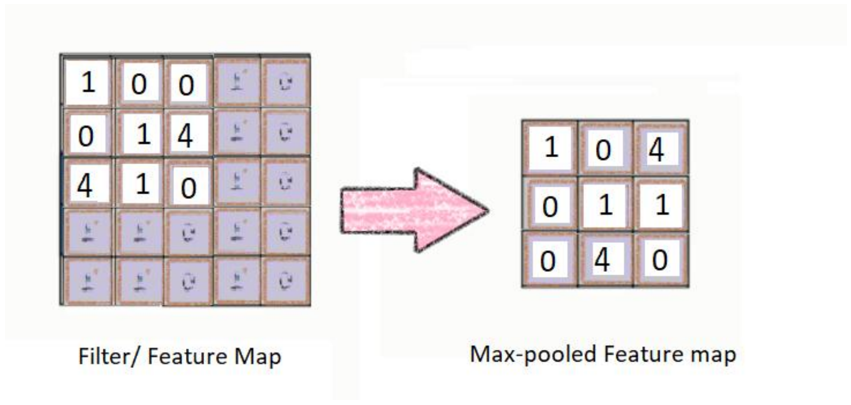
● Feature map: contains the arguments recorded by the filter over the input image. The value from the input image is initially inserted in the top-left cell of the feature map, and it moves a block to the right, recording the observation of every stride.



Step 2- Max Pooling:

The purpose of max-pooling is to enable the CNN to detect an image when presented with basic modification (flipped, mirrored, upside-down). In this step, we determine a pooled feature map. This process involves placing a smaller matrix on top of the feature map and insert the maximum value within the matrix into our pooled feature map. Then we keep moving towards the right until we fill all the values, as it was done in the convolution step.

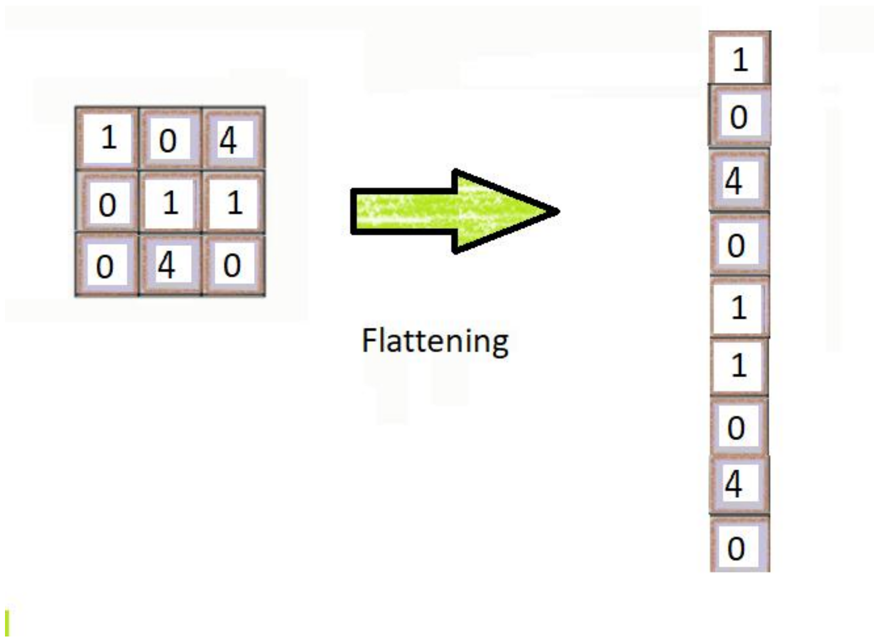
● Max Pooling is concerned with teaching the convolutional neural network to recognize that despite all of these differences, they are all images of the same thing. In order to do that, the network needs to acquire a property that is known as “spatial variance.”, so it can recognize an object in an image even if it is spatially different from another image of the same object. There are also other pooling techniques such as Mean Pooling (Takes the average), and Sum Pooling (Takes the Sum).



<u>Step 3 — Flattening:</u>

By the time we reach this step, we have a pooled feature map by now. As the name of this step implies, we are literally going to flatten our pooled feature map into the shape of a column. So instead of looking like a box squared matrix, our pooled feature map now looks like a vertical column.

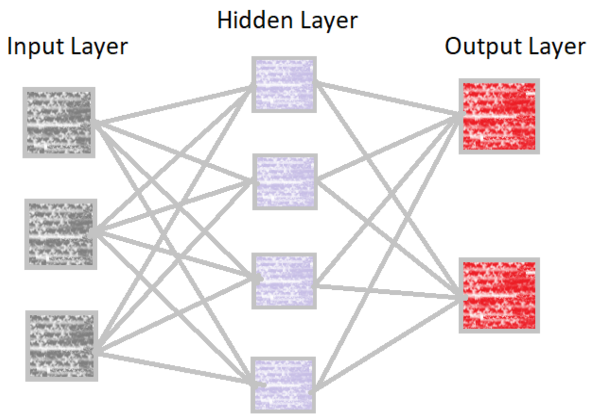
After flattening is that we end up with a long vector of input data that we then pass through the artificial neural network to have it processed further. If we didn't do this step, it would be hard for the Neural Network to read our data.



<u>Step 4 - Full Connection:</u>

The fully connected layer in the CNN is the same as a hidden layer in an ANN. The role of the artificial neural network is to take this data and combine the features into a wider variety of attributes that make the convolutional network more capable of classifying images.

● This also the step where we calculate the error function that our network takes into account before making predictions. In an ANN, it was called the Loss Function. The machine can now place weights on each of the fully-connected layers to determine the binary outcome of our independent variable.



Full Connection

Chapter 6

Results

First we built our model using CNN and trained it using a subset of images from the MNIST-ham10000 dataset. We then tested the said model using the remaining images from the same dataset. After plotting various graphs we found that Loss is around 50 percent and the Accuracy at which our model can predict the type of cancer correctly is around 60 percent initially but increases to almost 72 percent after subsequent run.

Conclusion

In this paper we have discussed a computer-aided diagnosis system for melanoma skin cancer. It can be concluded from the results that the proposed system can be effectively used by patients and physicians to diagnose the skin cancer more accurately. This tool is more useful for the rural areas where the experts in the medical field may not be available. Since the tool is made more user friendly and robust for images acquired in any conditions, it can serve the purpose of automatic diagnostics of the Skin Cancer.

Reference

1. American Cancer Society, “Cancer Facts & Figures 2016,” American Cancer Society, Atlanta, GA, USA, 2016.
2. Skin Cancer Facts and Statistics [Online]. Available: www.skincancer.org
3. Cancer.org
4. Kaggle.com for the dataset of images
5. Articles by Dr, Rohan gupta on Deep Learning
6. Medium.com for understanding better concept
7. Google.com for more extra informations