**Pothole Detection using Deep Learning techniques**

## Introduction

Road potholes are developed as a result of roadway wearing and damage and road corrosion. Individuals not only experience pain and suffering when passing through these potholes, but they also die as a result of road accidents. In the United States, about 2,000 people die every year due to poor road conditions or potholes. The Indian government, between 2013 and 2016, has reported that potholes caused 11,836 deaths and 36,421 injuries. Since the streets are submerged with water each year due to catastrophes, floods, and severe rains, pothole issues cannot be addressed simply.

Road evaluation and repair necessitates relationships among local citizens, public projects agencies, and private construction companies. Despite our inability to maintain the road, we are able to help lower the number of accidents that occur every year. To address the problem of potholes on the roads, I plan to use the Pothole detection dataset that contains various images of potholes and normal roads and build an efficient Deep Learning model that could detect the potholes on the road. Since this project is based on image processing using deep learning which is a part of Artificial Intelligence. This project on detecting potholes can be implemented in self-driving cars that run on Artificial Intelligence to avoid road accidents hence it makes it safer for people to travel in the vehicles. The first stage toward timely road surface repair is to determine the state of the road it terms of faults. Potholes are the most common problem on road surfaces. bowl-shaped holes in the road surface of varying diameters Potholes arise on the road surface due to a variety of factors. Wear and tear on the road, as well as the age of both the road, are all factors to consider. heavy rainfall, materials' climate-change resiliency external variables such as the use of poor construction materials and external elements such as a clogged sewer and bad construction management Recently, Roads with potholes, as depicted in, are damaged expanding in India, particularly in Mumbai, and as a result, complaints Pothole-related incidents are also on the rise. a decent route The state has traditionally contributed a significant share of the country's GDP economy.

## Dataset

The data that I will be using in this project is secondary data that is readily available in Kaggle. The dataset consists of two files one file includes the images that belong to normal images and the other file consists of the potholes images. There are a total of 681 scanned images in the dataset, which is quite a smaller number of images for training a deep learning model. Hence, we can use data augmentation to generate new images from the existing data and increase the number of images in the dataset. I have downloaded the dataset from Kaggle <https://www.kaggle.com/atulyakumar98/pothole-detection-dataset?select=potholes>.This dataset contains a balanced dataset of images.

## 1.2 Project Aim

In this project our aim is to detect the potholes using transfer learning by training these networks with different architectures and evaluate the networks based on the accuracy they have achieved. Our primary aim remains to build a classification model that could classify between the potholes.

## 1.3. Research Questions

The construction of research questions is the key part of any project. This is because the project or the research tries to answer these questions and come up with conclusions.

* What is the suitable architecture for Transfer learning model to achieve the best accuracy score?
* What is the role of Artificial Intelligence and Deep Learning in the self-driving cars for the purpose of object detections and similar topics?

# 1.4. Objectives

* Conduct study on the CNN classifiers and understand the function in depth.
* Use the knowledge acquired in the previous coursework’s and apply the learnings in the project.
* Train an image processing classifier(CNN) that could detect the potholes on the roads accurately.
* Experiment with different structures of the CNN to get the best output from the algorithm.
* Study the differences in the performances and understand the reasons behind the differences in the performance of the CNN models.
* Record the findings from the project and report them in the final report of the project with all the evidences and conclusions that were made during this project.

# 1.5. Tools and Techniques

The techniques that will be used in this project are Artificial Intelligence, Deep Learning, Image Processing, Computer Vision, Convolutional Neural Networks, Transfer Learning, and Data Visualization.

The tools that will be used in this project are Python programming language, Keras framework, Tensorflow, NumPy, Sklearn, etc.

There are several tools employed as phase of this project to achieve its main objective. a few essential tools utilized during this Endeavour include:

Tensorflow : Tensorflow is a machine learning and ai - based software library that is free and open-source. It can be used for a variety of applications, but it focuses on deep neural network training and inference. Tensorflow is a dataflow and differentiable programming-based symbolic math toolkit.

NumPy: NumPy is a Library in python designed to help with array management. It has features for linear algebra, Fourier transforms, and matrices, as well.

Pandas: Pandas is a Python-based library that works with data analysis and manipulation. The focus is on the operations and data structures required to manipulate tables and time - series data.

Scikit-learn: Scikit-learn is a library for Python that helps with machine learning. It is open source and available to anyone. It includes several algorithms, such as support vector machines, among others.

Matplotlib: Matplotlib is a tremendous 2D plotting library in Python, perfect for visualizing array data. Matplotlib is a library built on the concept of NumPy arrays, and it is made to work with the other components of the Scipy stack. Matplotlib has numerous plots, including lines, bars, scatter plots, histograms, and more.

Seaborn: Seaborn is an example of a Python library that works with the matplotlib data visualization framework and integrates with pandas data structures. Seaborn is Seaborn's central visualization system, which is crucial in helping the exploration of data. See how the distribution is univariate and bivariate.

## 1.6. Ethical, Legal, and Social issues

# Ethical Issues

After referring to the Ethical OS Toolkit, I have identified the possible risk zones that the project might fall into. The risk zones are explained below.

Machine Ethics & Algorithmic Biases

The algorithms have high chances of being biased if the data that is fed into the network is improper, or imbalanced, i.e., one class dominating the other class in the data. As a result of this, the algorithms behave biased towards the class that has the majority of the data points and misclassify the minority class in the data. This will show effects on the real-world scenarios when the model is used in production. The algorithm starts to misclassify the points, and this might end up in not identifying the potholes accurately.

Economic & Asset Inequalities

Due to the automation of the potholes detection on the roadways, this could show an impact on certain job roles which might be in threat of losing jobs. Though automation is necessary it is also highly important to make sure that job losses do not occur in this process. The employees work hard and are loyal to the organizations and expect the organizations to be loyal to them Axing the employees because of the automation process could be ethically wrong. Instead, the employees should be skilled by the organizations on the new technologies and use their resources efficiently.

# Legal Issues

Legal issues are the ones that occur with the breach of laws set by the government. The automation may sometimes fail in detecting the potholes depending on the different light conditions and this might end up in not filling a few potholes that may result in road accidents. Legal actions will be taken on certain individuals who are responsible for the potholes on the roads.

# Social Issues

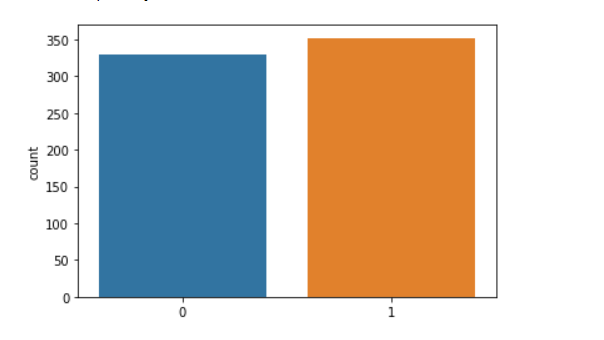
The political issues might occur in some areas while predicting and repairing the potholes. This usually happens in countries that are underdeveloped. These political issues could cause certain problems in the pothole’s detection and repairs. Hence it is recommended to take all the necessary permissions from the local politicians.

## 2. Methodology

A detailed explanation of the technique that was utilized to effectively accomplish this project is provided in this section of the report. I started by installing all of the necessary software, such as Python and Anaconda, and then I went on to install all of the libraries that were needed via the command prompt. NumPy, OS, CV2, Matplotlib, SciKit Learn, and Keras are some of the libraries that were used in the course of the research. The data was obtained via the Kaggle website.. There are two kinds of files in this collection: the normal class and the potholes class.

## 2.1 Importing Data

We are working on Jupyter environment since it has a very interactive interface, and it is suitable for production. This IDE is the most used by Data Scientists across the globe. The initial step was to import all the libraries in the notebook. Firstly, we have to import all the data that was collected from the Kaggle in order to proceed with our research. All the data will be imported to the Jupyter IDE. I have used the OS library to read the directory of the data file location and I have imported the dataset and assigned them to train and test variables according to the way they were organized. Since the whole data was divided into two classes, I have plotted a bar plot to see how the data is distributed among the classes in the train data.



Here 0 indicates normal class and 1 indicates potholes class.

## 2.2 Data Visualization

In Data Visualization I have visualized the images of potholes and normal roads.





## 2.3. Image Pre-processing

After importing the data into the IDE, I have separated the labels and features and stored them in different variables for all the sets, i.e., train and test. We further perform the image pre-processing on the data to normalize and re-size the images. This step is intended to enhance the picture quality so where we can conduct a more thorough analysis of the image. The use of pre-processing allows us to eliminate unwanted distortions and improve certain characteristics that are essential for the specific application for which we are developing. Those characteristics may vary depending on the application.

There are two pre-processing stems that were performed as discussed. They are as follows:

### 

## 2.3.1. Normalization:

Data normalization is a crucial step in the image processing process since it guarantees that every input parameter (in this instance, pixel) has a comparable data distribution. As a result, the network's convergence time is reduced while it is being trained. Data normalization is accomplished by removing the average out of each pixel and dividing the resulting value by a standard deviation of the data set. In this case, the distribution of the data may resemble a Gaussian distribution with the origin at zero. We need positive pixel values for picture inputs, therefore we could choose to scale the normalized data in the range [0, 1] or [0, 255], depending on the application.

For the purpose of normalizing our data, I have transformed the features into a NumPy array and then divided the array by 255. In the train, validation, and test sets, this process was repeated for each feature set.

## 2.3.2. Resizing images

In computer vision, resizing pictures is a crucial step in the pre-processing process. Machine learning techniques learn to recognize patterns in smaller pictures, which is important since they train quicker on smaller images. The learning process for a twice-larger input images needs the network to learn over 4 times as much of pixels, which increases the amount of time it takes. A further point to consider is that many neural network model designs need input images to be of the same size, while our raw gathered images may be of different sizes.

It was necessary to scale the whole data set to fall inside the range of [-1, 1] across all of the feature sets. This ensures that the image's size falls within the boundaries of the specified range.

## 2.5 Deep Learning

Artificial Intelligence has seen tremendous progress in recent years in terms of reducing the gap in between abilities of mankind and those of computers. Researchers are working on a variety of areas of the field in order to bring about amazing results. In this regard, the field of Computer Vision is only one among several. Among the objectives of this field is to enable machines to view and comprehend their environment in the same way that humans do, and to use this expertise for a variety of tasks such as image and video recognition, image processing & categorization , recommendation systems, etc. The advances in Computer Vision in regard with Neural Networks have been built and refined over time, mainly via the use of a single algorithm — the Convolutional Neural Network as the foundation. Deep learning technique that allows computers to learn by example in the same way that humans do. Deep learning is a critical component of self-driving automobiles, allowing them to detect a stop sign or discriminate between a pedestrian and a lamppost. It enables voice commands in consumer electronics such as phones, tablets, televisions, and hands-free speakers. Deep learning has gotten a lot of press recently, and with good cause. It's accomplishing accomplishments that were previously unattainable. A computers model learns to execute categorization tasks directly from photos, text, or sounds in deep learning. Deep learning models can attain state-of-the-art accuracy, even surpassing human performance in some cases. Models are trained utilizing a huge quantity of labelled data and multilayer neural network topologies.

When given an images as input, a Convolutional Neural Network (ConvNet) may assign significance to various features or objects in the image, and then distinguish between them. When contrasted to all other classification methods, the amount of pre-processing needed by a ConvNet is much less. While basic techniques need filters to be hand-engineered, ConvNets have the capability of learning these characteristics with sufficient training. In many ways, the design of a ConvNet is similar to the connection patterns of Neurons in the Structure Of the brain, and it was influenced by the structure of the Visual Cortex. Every single neuron in the human visual system responds to signals primarily in a certain area of the field of vision termed as Receptive Field. A group of similar fields may be used to fill the whole visual region by overlapping them. (Saha, 2018)

## 2.5.1. Architecture of CNN

Every layer of a basic ConvNet converts one region of activation functions to another via the use of a differentiable function, and each layer of a simple ConvNet is composed of layers. Convolutional Network designs are built using three kinds of layers: the Convolutional Layer, the Pooling Layer, and the Fully-Connected Layer. These layers will be stacked together to create a complete ConvNet architecture.



Figure 5 - Architecture of CNN (source: http://cs231n.stanford.edu/slides/2017/cs231n\_2017\_lecture6.pdf)

As mentioned above there are various layers in the architecture of a CNN. The layers can be described as follows:

1. Input Layer: The raw image pixels of the picture will be passed through the input layer.
2. Convolutional Layer: Convolution layer is the most essential building blocks of convolutional neural networks, and they are utilized in many different applications. Convolutional layers operate on the input by applying a convolution operation and transferring the output towards the next layer. A convolution is a mathematical operation that transforms all of the pixels through its receiving area it in to a numeric measure. For instance, if we perform a convolution to a picture, we will be reducing the size of the image while also condensing all of the data in the field into single pixel in the process. A vector is produced as the final outcome of the convolutional layer. Different types of convolutions may be used depending on the kind of issue we are attempting to solve and the type of the attributes we are attempting to learn from the data.
3. Pooling Layer: Pooling layers are a technique for down sample feature maps that summarizes the existence of features in regions of the feature map, allowing for more efficient sampling. Average pooling & max pooling are both often used pooling techniques that summaries the mean existence of a feature as well as the most active presence of a feature, respectively, in a dataset.
4. Fully connected Layer: Feed forward networks are what the Fully Connected Layer is all about. The Fully Connected Levels are the last few layers of the network hierarchy. A flattened version of the outputs from the last Pooling Layer is sent into the fully connected layer, which is then fed back into the fully connected layer.

**Activation Function:**

Here in these networks, the activation function used was ReLu activation function. the rectified linear activation function also known as ReLu is a non-linear function that could produce the input directly if the input is positive and will otherwise produce zero if the input is negative. Due to the fact that a model that employs it is simpler to train and produces higher performance in many cases.

Chart, line chart, scatter chart

Description automatically generated

Figure 6 - ReLu activation function (Toprak, 2020)

**Loss Function:**

Binary cross entropy loss function was used in the CNN network since this is a two class classification problem. The binary cross entropy is calculated using the following formula:

Logo, company name

Description automatically generated

Figure 7 - Binary cross entropy loss function

**Optimizer:**

The optimizer used in the CNN network is ADAM optimizer, also known as Adaptive Moment Estimation, is an optimization method for gradient descent that is based on an algorithm. When dealing with a massive problem containing a large number of data points or parameters, this technique is very efficient. As a result, it uses less memory and is more efficient.

**Evaluation Metrics:**

The evaluation metrics used to evaluate the CNN models is Accuracy. On the basis of training data, accuracy is the metric used to evaluate which model is the most effective at detecting patterns and connections among variables in a dataset.

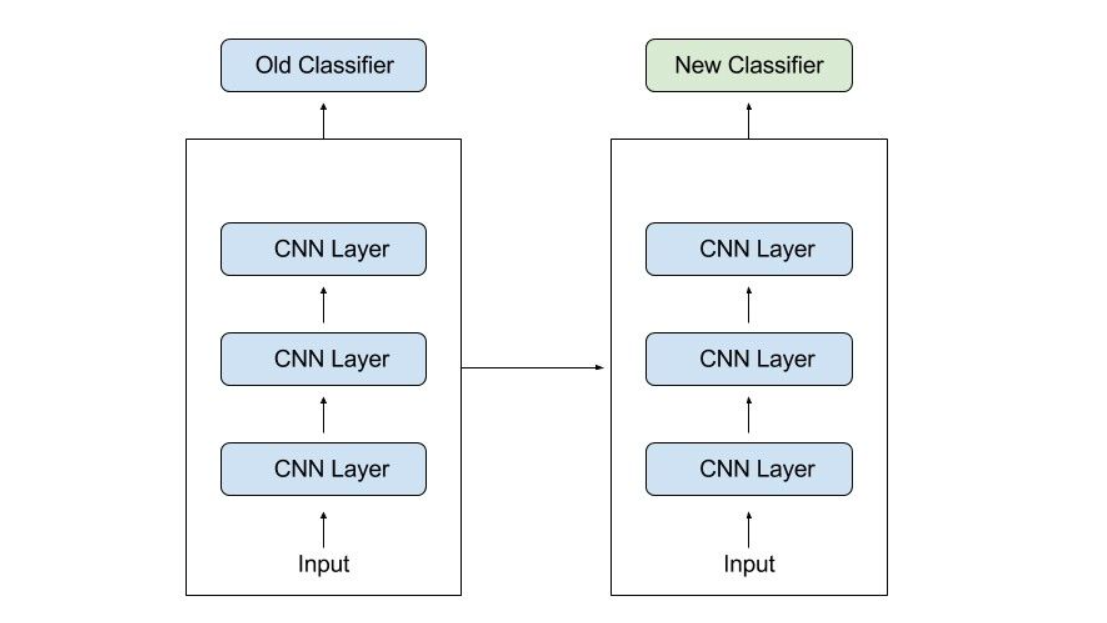
## 2.6 Transfer Learning

Transforming a model is trained on a huge database to another smaller one is at the heart of transfer learning. To use a CNN for object recognition, we freeze the network's early convolutional layers and train only the final few levels that make predictions. A pre-trained model is reused on a new problem through transfer learning (TL). Due to its ability to train convolutional neural networks with relatively little input, it's now quite popular among deep learning researchers. Since most real-world situations don't have millions of labelled data sets to train such complicated models, this is extremely helpful inside the data science field. Our goal is to provide you a better understanding of transfer learning. Transfer learning will be discussed, as well as some resources on pre-trained models will be provided.

## 2.6.1 Working of Transfer Learning

Brain networks are often used in computer vision to recognize edges, forms and task-specific properties in the early levels of the computer's vision system. The early to mid layers are utilized in transfer learning, and only the latter layers are retrained. It makes use of the labelled data from the task this was originally trained on.

Consider the case of a model that was trained to recognize a backpack in a picture and be used to detect sunglasses. For this reason, only the last layers will be retrained to learn how sunglasses differ from other objects. The model has already learned to distinguish objects in the prior levels.



As much information as feasible from the previous work is transferred to the current task when using transfer learning. Information can take a variety of shapes based on the situation and the available data. For example, the way models are put together may help us detect new items more quickly.

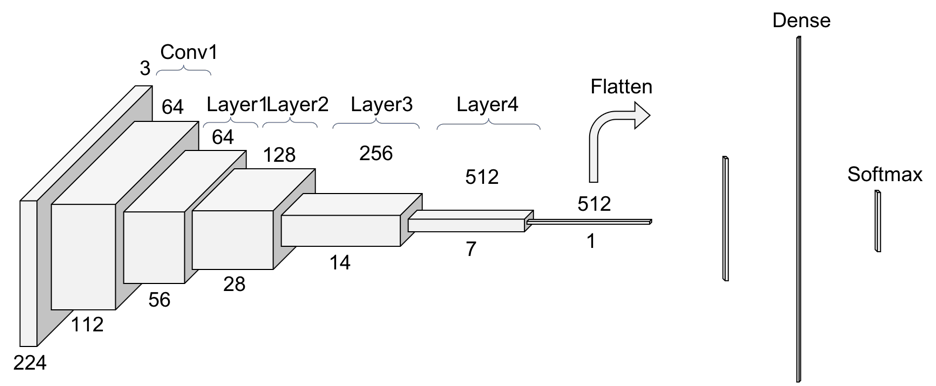
In this project I have implemented two models of transfer learning.

## 2.6.2. Resnet v2

It differs from regular sequential networks in that ResNet uses microarchitecture modules known as a network inside the architecture rather than sequential network such as OverFeat and VGG. New networks are often built from the ground up using microarchitecture, which is a collection of small building blocks. Based on Kaiming He's seminal work that was first published in 2015, this research shows how deep network can be train using regular SGD by modifying the residue module to employ identity mapping as proven in the previous work.

In spite of ResNet's greater depth, the real model weights are smaller because average global pooling instead of the convolution layer is employed, resulting in a smaller model size for ResNet50, which is more commonly used for 100MB's models.

## Architecture :



These are the some important points of Resnet v2.

* It is based on the 'Residual Learning' idea. Residual Learning that is based on stacking residual blocks to improve performance of neural networks.
* Resnet stands for Residual Network, which is a network that facilitates Residual Learning. The number 50 denotes the number of layers. Resnet50 refers to a 50-layer residual network.
* Due to their inability to extract significant information from images, simpler Neural networks do not work well on dataset.
* As a result, the number of hidden layers will automatically grow in order to improve performance indicators like as accuracy and AUROC. However, when we add additional layers to our neural networks, the accuracy begins to saturate and ultimately decline. Residual Training aims to address this issue.
* Instead of attempting to learn some attributes, residual learning focuses on learning some residual. Residual could be simply defined as the removal of a feature learned from a layer's input. Shortcut connection (directly connecting the input of the nth layer to the input of some (n+x)th layer) are used by ResNet to accomplish this. It has been demonstrated that training this type of network is simpler than training simple cnn model, and that the issue of accuracy degradation has been handled.
* Inputs can propagate quicker through back propagation across layers with residual blocks, and Batch Normalization layers are used to speed it up runtime and prevent overfitting.
* The Vanishing Gradients issue is also addressed by ResNets. It prevents the gradients from rapidly decreasing to zero. A "Quick" or a Skip Connections in ResNet allows a gradients to be directly back-propagated to previous layers.

The Implementation of Resnet V2 for this project is :



Training loss : 0.069 Testing loss : 1.58

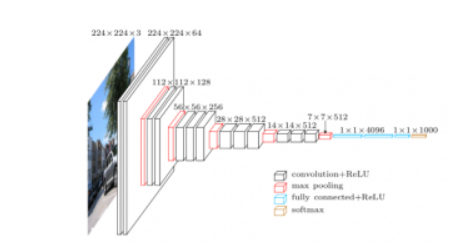
Training Accuracy: 97.98% Testing Accuracy : 81.02%

## 2.6.3. VGGNet16

Karen Simonyan and Andrew Zisserman of the University of Oxford proposed the VGGNet architecture for a Convolutional Neural Network in 2014. The main focus of this research is on the accuracy of the cnn model as a function of its depth. Very Convolutional Neural Network for Wide Scale Image Processing is the title of the original study on VGGNet.

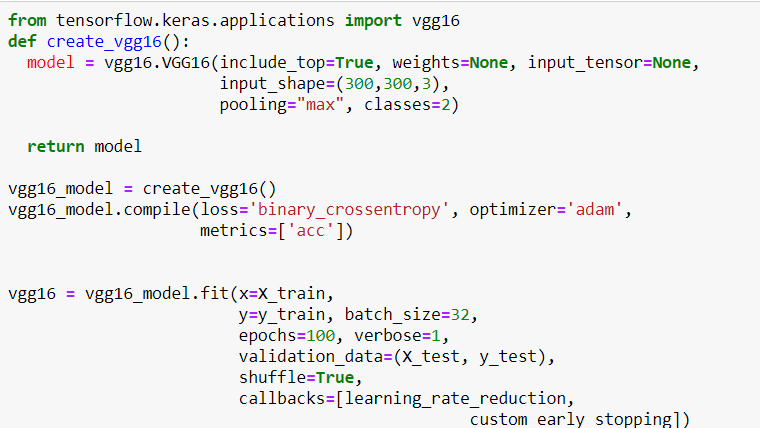
## VGGNet Architecture

When using VGG-based convNet, you'll need a 224\*224 Rgb as your input. The preprocessing layer subtracts the average image values determined for the complete ImageNet training set from the Image with pixels values ranging from 0–255.



These weight layers are applied to the preprocessed input photos. A series of convolutional layers is applied to the practice images. In the VGG16 architecture, there are a total of Thirteen convolution layer and three fully connected layers. Instead of using huge filters, VGG uses smaller ones (3\*3) that have more depth in their image processing capabilities. Using this method, the receptive field is identical to that of using only one 7 x 7 convolution layers.

The Implementation of VGG16 for this project is :

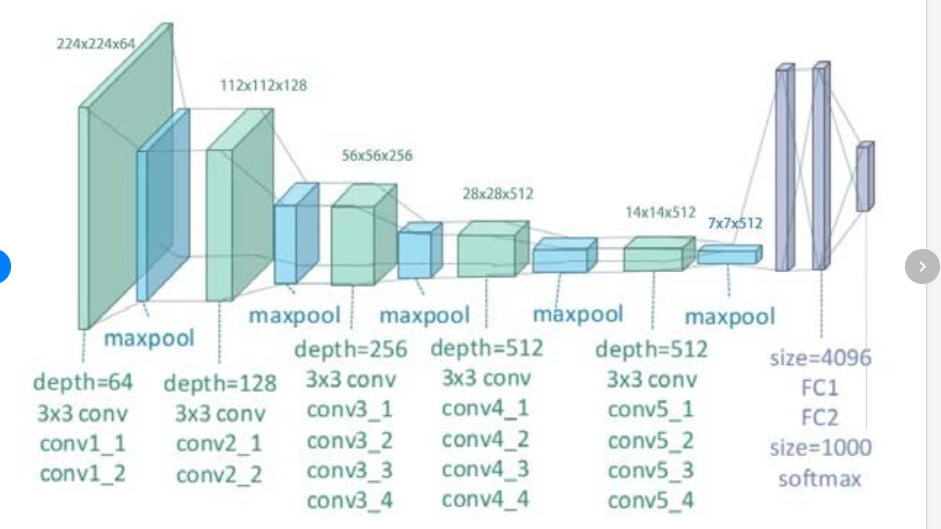


Training loss : 0.071 Testing loss : 0.5030

Training Accuracy: 98% Testing Accuracy : 89.01%

## 2.6.4 VGG 19

VGG-19 is a 19-layer deep convolutional neural network. You can import a pretrained form of the network from the ImageNet dataset , which has been trained on over a million photos. The network can classify photos into 1000 different object categories, including keyboards, mice, pencils, and a variety of animals.



Architecture of VGG19

Training loss : 0.9667 Testing loss : 0.4943

Training Accuracy: 94.71% Testing Accuracy : 84.71%

## 2.7. Evaluation

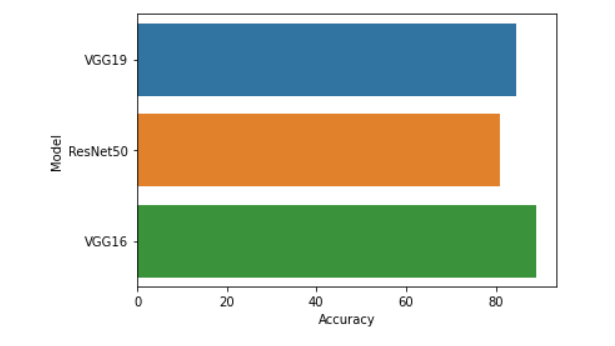
The evaluation metrics used to evaluate the CNN models is Accuracy. On the basis of training data, accuracy is the metric used to evaluate which model is the most effective at detecting patterns and connections among variables in a dataset.

After training the models on the training set and validation on the validation set using the accuracy as metric, I have further evaluated the models on the test set that was provided along with the dataset.

The following table describes the accuracy scores and the loss for each model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Train Accuracy** | **Train Loss** | **Test Accuracy** | **Test Loss** |
| Resnet 50 | 97.98% | 0.069 | 81.02% | 1.25 |
| VGG 16 | 98% | 0.0701 | 89.01% | 0.5030 |
| VGG 19 | 94.51% | 0.9667 | 84.71% | 0.4943 |

Bar graph of Accuracy



## 3. Conclusion

Road evaluation and repair necessitates relationships among local citizens, public projects agencies, and private construction companies. Despite our inability to maintain the road, we are able to help lower the number of accidents that occur every year. To address the problem of potholes on the roads, I plan to use the Pothole detection dataset that contains various images of potholes and normal roads and build an efficient Deep Learning model that could detect the potholes on the road. In this project we are trying to predict the potholes using the image evidences that were provided in the dataset. Using this dataset, we have performed several pre-processing operations and trained transfer learning models with different architectures. Upon comparing the performances of all the models that were trained as part of this project it has been observed that the performance of VGG16 is more efficient on all the training and test sets. Various neural network strategies for potholes detection and classification were discussed in this systematic review research. These methods are all non-invasive. Preprocessing and picture segmentation are two phases in the detection of skin cancer. The focus of this review was on transfer learning for lesion picture classification. Each algorithm has its own set of benefits and drawbacks. The most important factor in achieving the best results is choosing the right classification technique. When it comes to classifying picture data, however, VGG16 outperforms other types of neural networks because it is more directly tied to computer vision than other types of neural networks.

Based on the analysis my findings from answering the research questions:

VGG16 gives higher accuracy as compare to other transfer learning models. The fundamental advantage of VGG16 over its predecessors is that it discovers essential traits without the need for human intervention. Given a large number of photos of cats and dogs, it can learn the key characteristics of each class on its own. With the many deep neural networks constructed, a considerable result was reached on the ImageNet Challenger, which is the most significant picture classification and segmentation challenge in the image analyzing area.

The car's I software connects to all of the devices and collects data from Street view and the video cameras inside. The AI uses deep learning to replicate human perception and decision-making processes and to regulate actions in driving control systems such as steering and braking.