Training and Internship

On

Applied Machine Learning and Data Science

(Course Code: 002)

Project Report

On

Image Classification Project

Prepared for:
Indian Institute of Technology, Kanpur

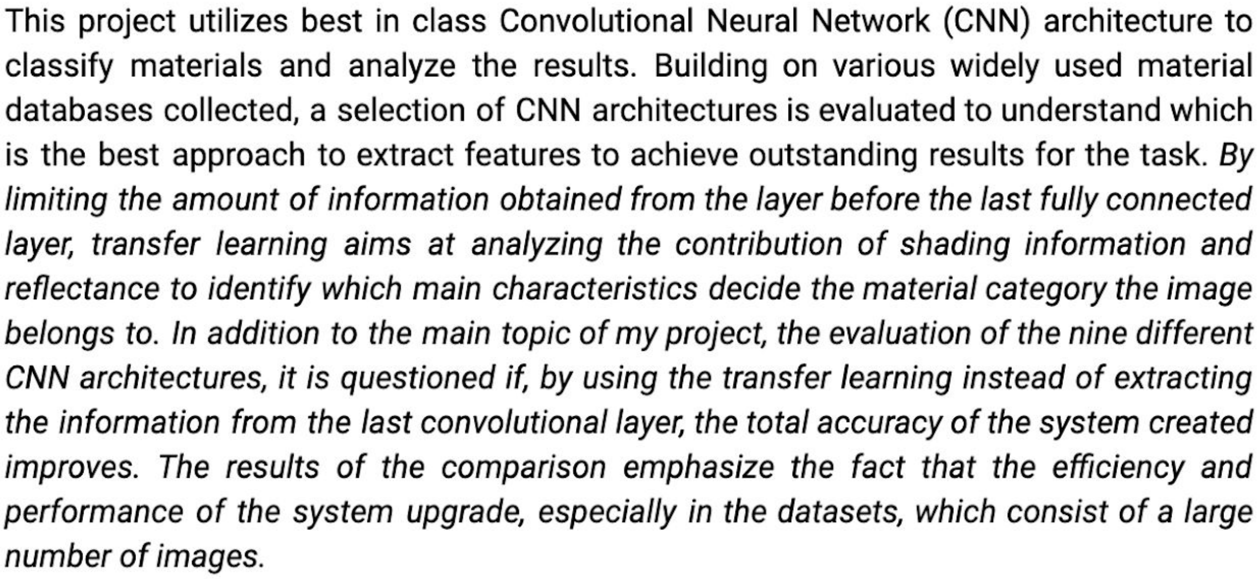
Submitted by:
Akshat Chand
Sneha Dahiya
Tanya Gupta
Ankita Singh
Siddharth Singh

Team Name:
Brute **F**orce

Abstract

The recognition and classification of the diversity of materials that exist in the environment around us are a critical visual competence that computer vision systems focus on in recent years. Understanding the identification of materials in clear images involves an in-depth process that has made usage of the latest progress in neural networks, which has brought the potential to train architectures to extract features for this challenging task.

Image Classification holds the potential for a wide range of uses and in various industries. Different businesses possess massive databases with visual content, which is challenging to manage and make use of and hence, may end up uncategorized and useless. Fortunately, Classification of images through machine learning is a crucial solution for such a problem. With a useful model for image classification, companies can easily organize and categorize their database as it allows for automatic Classification of images in large quantities, helping companies monetize their visual content without investing countless hours for manual sorting and tagging.



This project utilizes best in class Convolutional Neural Network (CNN) architecture to classify materials and analyze the results. Building on various widely used material databases collected, a selection of CNN architectures is evaluated to understand which is the best approach to extract features to achieve outstanding results for the task. *By limiting the amount of information obtained from the layer before the last fully connected layer, transfer learning aims at analyzing the contribution of shading information and reflectance to identify which main characteristics decide the material category the image belongs to. In addition to the main topic of my project, the evaluation of the nine different CNN architectures, it is questioned if, by using the transfer learning instead of extracting the information from the last convolutional layer, the total accuracy of the system created improves. The results of the comparison emphasize the fact that the efficiency and performance of the system upgrade, especially in the datasets, which consist of a large number of images.*

1. Introduction

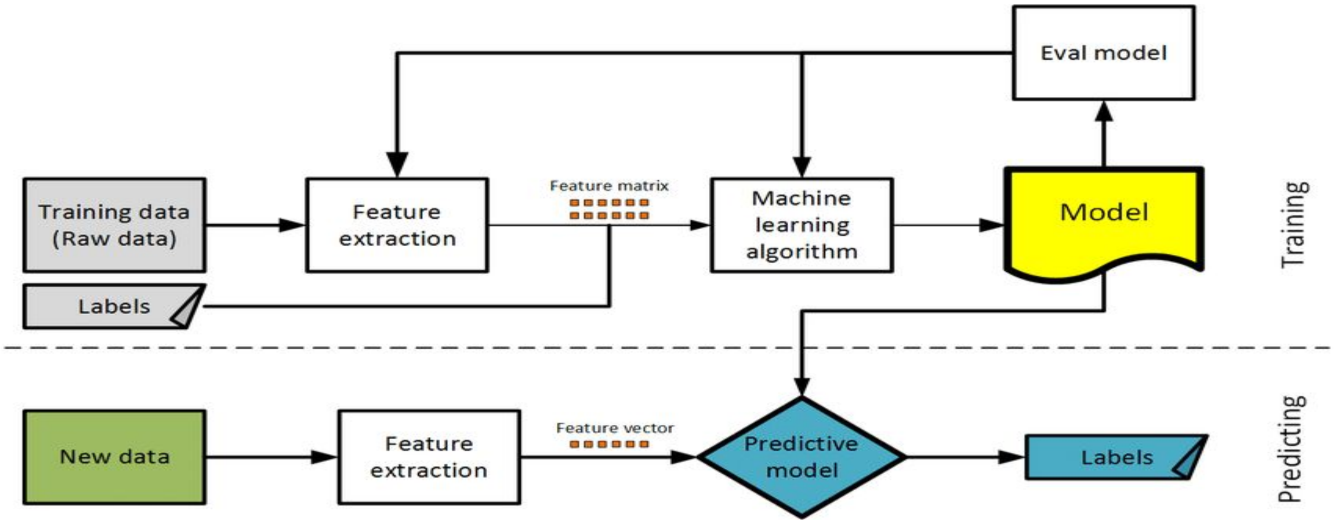
1.1. Motivation and Background

Computer vision is concerned with the extraction of meaningful information from image data. Projects on computer vision are often concerned with the development of computer algorithms for specific applications.
The Image Detect challenge from Kaggle has 200 classes, with each type having 450 training images, 50 validation images, and 50 test images.

1.2. Task Definition

The Image Detect challenge from Kaggle requires us to build the best model to classify the images provided in the dataset into respective classes.
The provided dataset consists of 200 classes. Each class has 450 training images, 50 validation images, and 50 test images. The training, validation, and test sets were provided with pictures and bounding boxes as annotations.
However, the task involved only to predict the class label of each image without localizing the objects. The test set consisted of no labels.

Our learning task is to learn a classification model to determine the decision boundary for the training dataset. The whole process is illustrated in Figure 1, from which we can see the input for the learning task is images from the training dataset, while the output is the learned classification model.



Our performance task is to apply the learned classification model to classify images from the test dataset, and then evaluate the classification accuracy. As seen from Figure 2, the input is images from the test dataset, and the output is the classification accuracy.

2. Methods

All models for this project were implemented in Python using the Keras library (v1.3) running on top of Theano(v.0.7).

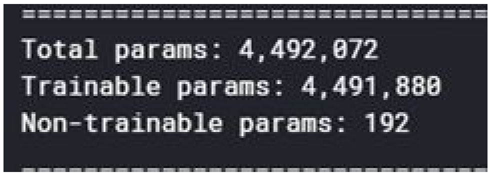
The image data were pre-processed by dividing all values by 255, to get values between (0,1), ImageDataGenerator was used for Data augmentation due to the limited size of training data.

2.1. Network Architecture Used

2.1.1. Model From Scratch

We first decided to create a very simple classifier. Our simple convolutional network consisted of six convolutional layer (each with ReLU activation) and two dense layers (one with ReLU activation and other with softmax). Batch Normalisation was used after second and fourth layer and Maxpooling with a pool size of 2x2 after every 2 layers. Our convolutional layer had a filter size of 5x5 for the first two layers and a filter size of 3 x 3 for the rest 4 layers. 32 filters were applied for the first two layers, 64 filters for the next 3 , and 128 filters for the last convolutional layer, and we used the same padding. We used categorical cross-entropy loss and Adam optimizer with a learning rate of 0.001. In addition, we used batch sizes of 128 images and l2 regularization.

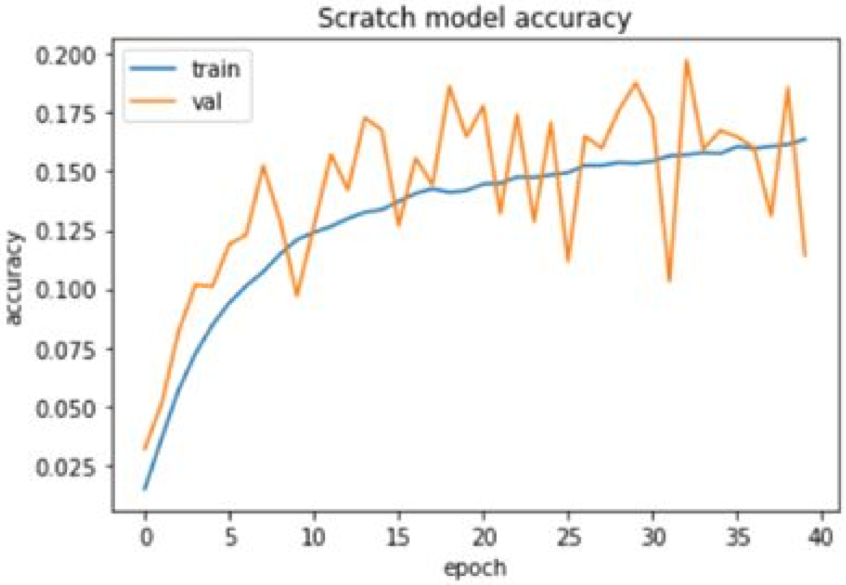
Dropout of 0.5 was also used after the first dense layer.

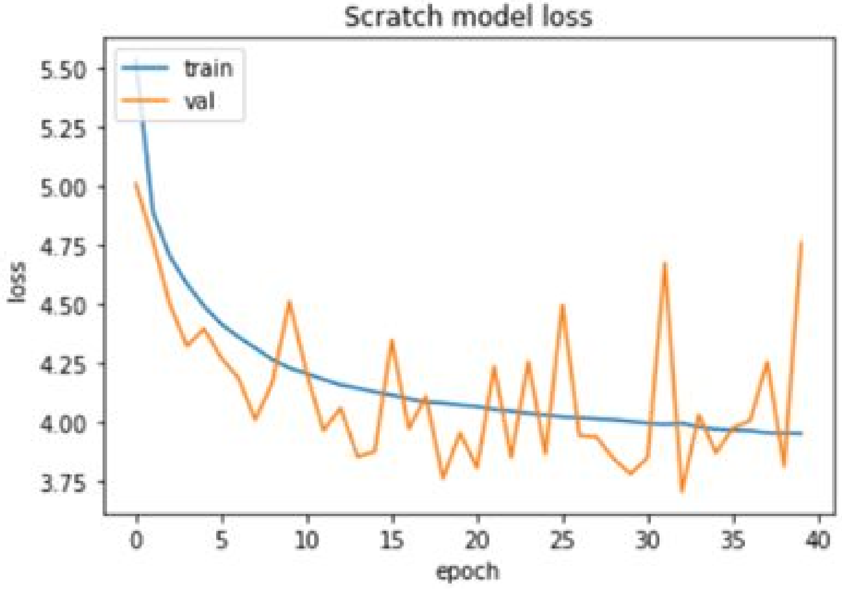


```
===============================
Total params: 4,492,072
Trainable params: 4,491,880
Non-trainable params: 192
_______________________________
```

Our simple convolutional network was able to achieve a validation accuracy of 0.1857 (figure) after 40 epochs.

Moving on to more complex models was the interpretation we could draw from scratch model.

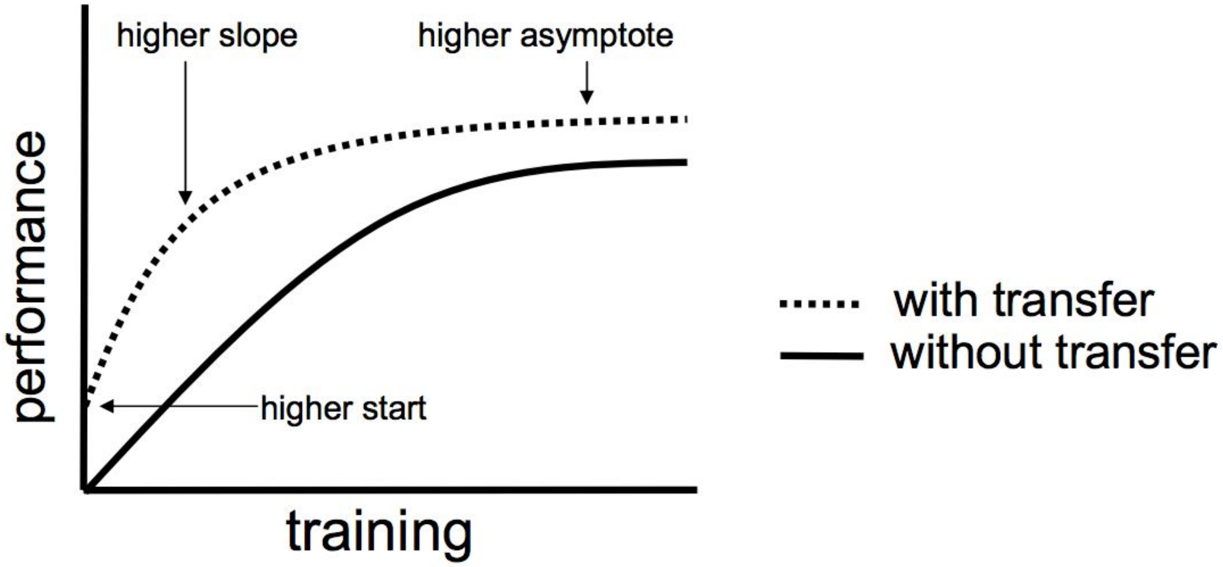

Scratch model accuracy
train
val
accuracy
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0
5
10
15
20
25
30
35
40
epoch


Scratch model loss
train
val
loss
5.50
5.25
5.00
4.75
4.50
4.25
4.00
3.75
0
5
10
15
20
25
30
35
40
epoch

2.1.2 Transfer Learning

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in the skill that they provide on related problems.



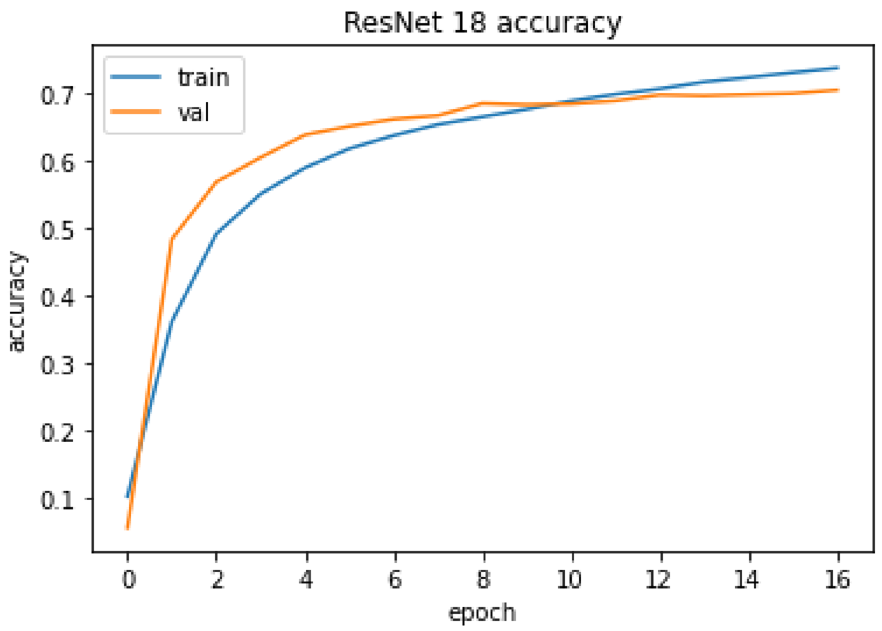
We tried following models for transfer learning.

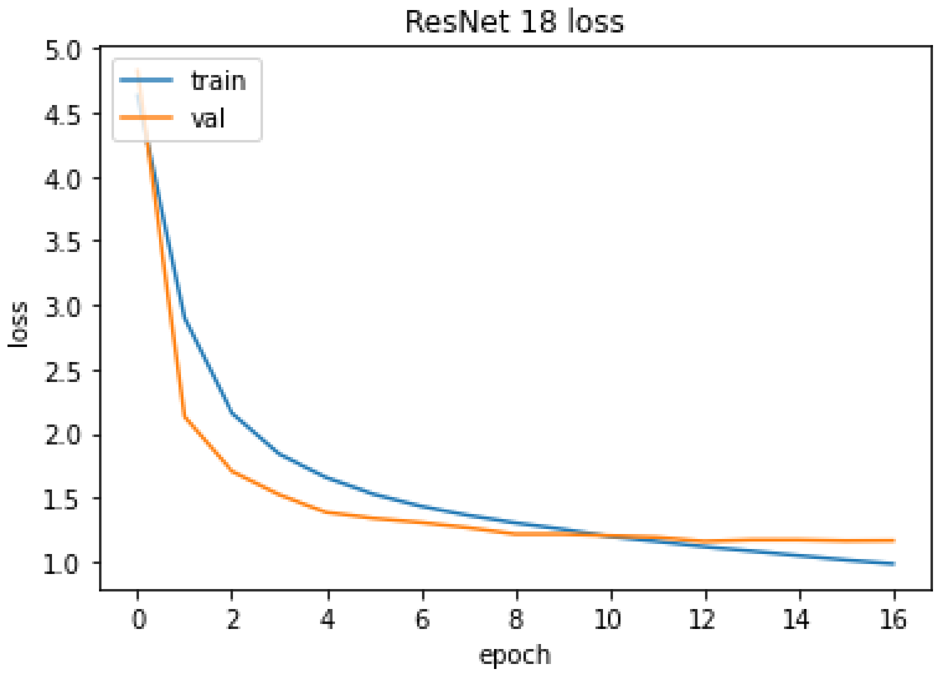
2.1.2.1 ResNet Neural Network

Residual Network (ResNet) is a Convolutional Neural Network (CNN) architecture that was designed to enable hundreds or thousands of convolutional layers. While previous CNN architectures had a drop off in the effectiveness of additional layers, ResNet can add a large number of layers with reliable performance.ResNet was an innovative solution to the “vanishing gradient” problem. Neural networks train via the backpropagation process, which relies on gradient descent, moving down the loss function to find the weights that minimize it. If there are too many layers, repeated multiplication makes the gradient smaller and smaller, until it “disappears,” causing performance to saturate or even degrade with each additional layer.

ResNets of varying depth were tried, it was noted that there was little to no advantage gained in accuracy upon increasing the intensity of the Network, it was clear that the problem set was suitable for a smaller network.

The final architecture which gave us the best results was pre-trained ResNet18 which was loaded without the last layer, after using GlobalAveragePool the output was made to be a softmax layer of 200 neurons to fit the problem set. Input image sizes of (64,64), randomcrop (56,56), and (224,224) were tried, and the best results were achieved when the input was (224,224) sized images which is the original input shape of ResNet.

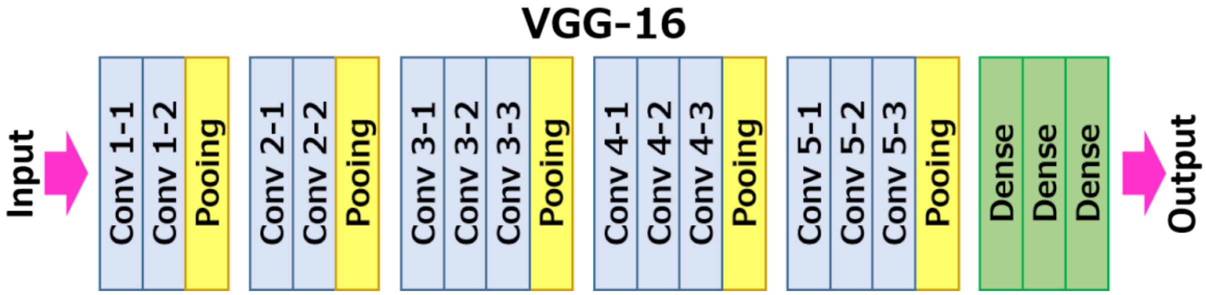


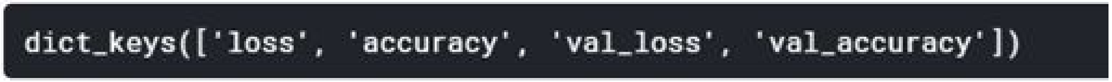


The ResNet solution is “identity shortcut connections.” ResNet stacks up identity mappings, layers that initially don’t do anything, and skips over them, reusing the activations from previous layers. Skipping initially compresses the network into only a few layers, which enables faster learning. Then, when the network trains again, all layers are expanded, and the “residual” parts of the network explore more and more of the feature space of the source image.

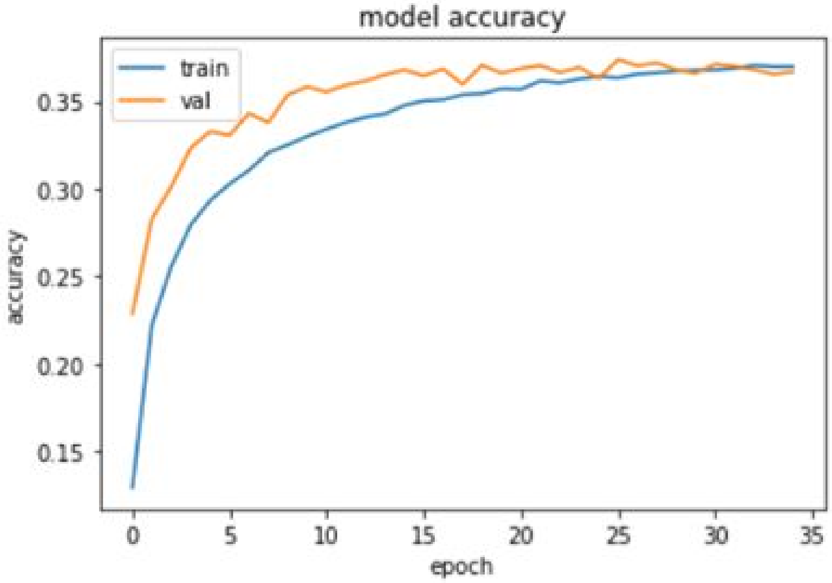
2.1.2.2. VGG-16 Convolutional Network

VGG 16 architecture was tried with Global Average Pool and two dense Fully connected layers of 1024 and 200 neurons. Due to a large number of parameters, training was prolonged. Upon freezing all layers and adding substantial dropout regularisation in the final layers the model accuracy stagnated at 0.35.





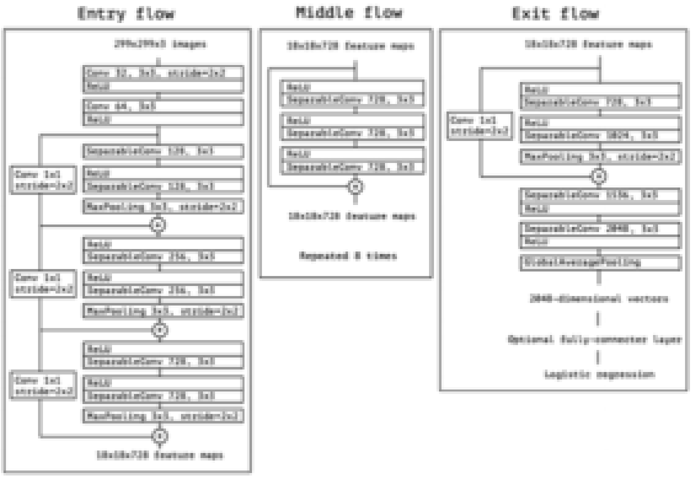
```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

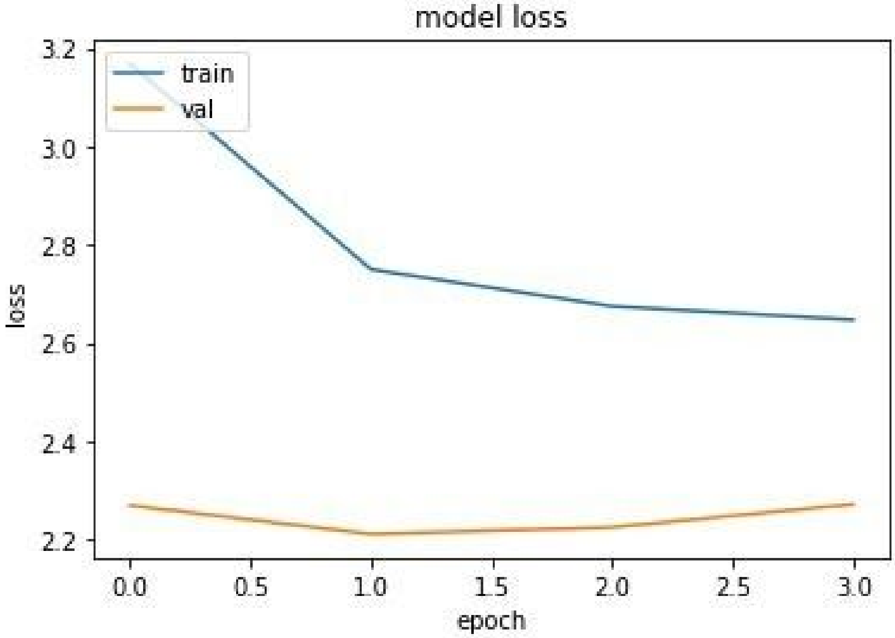


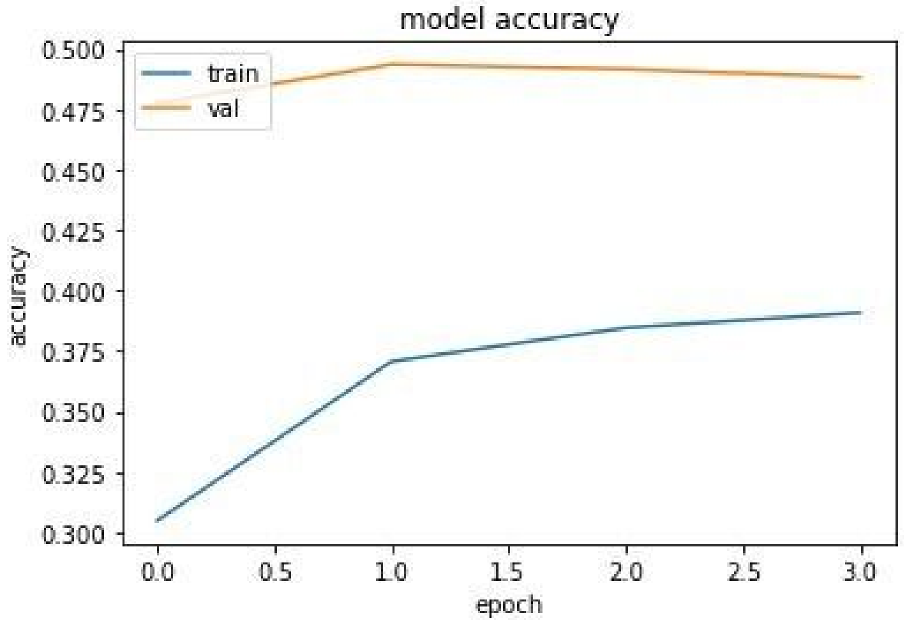
2.1.2.3. Xception Convolutional Network

Xception was proposed by none other than François Chollet himself, the creator and chief maintainer of the Keras library. Xception is an extension of the Inception architecture which replaces the standard Inception modules with depthwise separable convolutions

The image size fed into Xception was of the shape (128,128,3) as it was noticed that the accuracy was nominal for the original dimension. We are able to get an accuracy of 0.46, with only a few iterations on the training data. Further accuracy of 0.49 - 0.52 was achieved when the final 200 neuron layers of a simple CNN and Xception were added before making the prediction. This helped us understand that we were failing to extract particular features as a result of freezing the weights of the layers other than FC layers.


Entry flow
299x299x3 images
Conv 32, 3x3, stride=2x2
ReLU
Conv 64, 3x3
ReLU
SeparableConv 128, 3x3
Conv 1x1
stride=2x2
ReLU
SeparableConv 128, 3x3
MaxPooling 3x3, stride=2x2
ReLU
SeparableConv 256, 3x3
Conv 1x1
stride=2x2
ReLU
SeparableConv 256, 3x3
MaxPooling 3x3, stride=2x2
ReLU
SeparableConv 728, 3x3
Conv 1x1
stride=2x2
ReLU
SeparableConv 728, 3x3
MaxPooling 3x3, stride=2x2
19x19x728 feature maps
Middle flow
19x19x728 feature maps
ReLU
SeparableConv 728, 3x3
ReLU
SeparableConv 728, 3x3
ReLU
SeparableConv 728, 3x3
19x19x728 feature maps
Repeated 8 times
Exit flow
19x19x728 feature maps
ReLU
SeparableConv 728, 3x3
Conv 1x1
stride=2x2
ReLU
SeparableConv 1024, 3x3
MaxPooling 3x3, stride=2x2
SeparableConv 1536, 3x3
ReLU
SeparableConv 2048, 3x3
ReLU
GlobalAveragePooling
2048-dimensional vectors
Optional fully-connected layer
Logistic regression


model loss
train
val
loss
3.2
3.0
2.8
2.6
2.4
2.2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
epoch

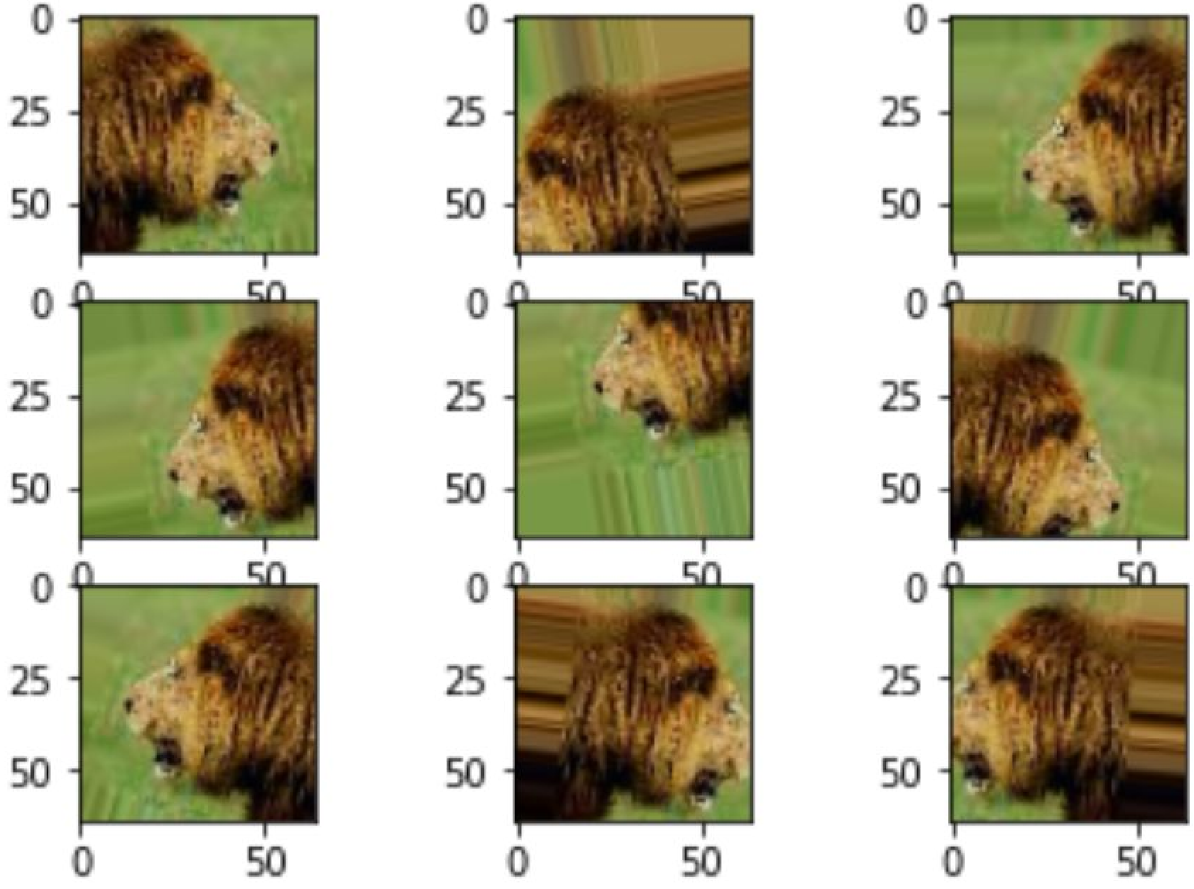


2.2. Optimization and Experimentation

2.2.1 Data Augmentation :

The dataset provided contains 450 images for training and 50 images for validation for each of its 200 labeled classes. Since the data is limited, data augmentation was applied using the Keras Image Data Generator functions. An example of the augmented images has been shown. The parameters were as follows:

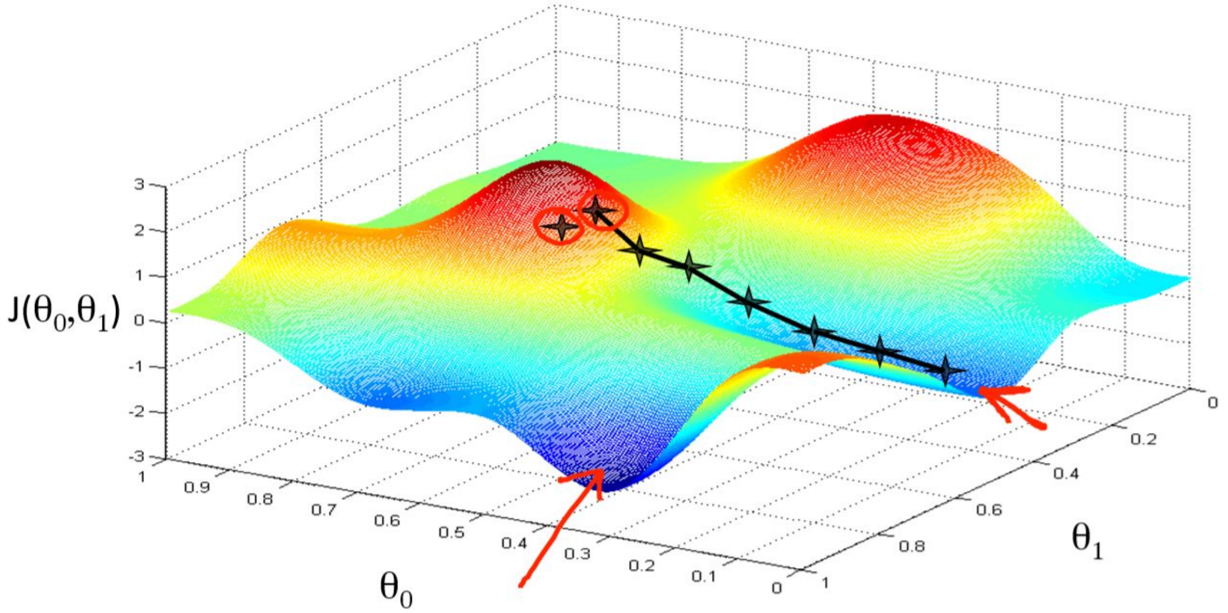
ImageDataGenerator(rescale=1./255,rotation_range=20,horizontal_flip=True ,width_shift_range = 0.3,height_shift_range=0.3)



2.2.2 Stochastic Gradient Descent:

“Gradient descent is an iterative algorithm, that starts from a random point on a function and travels down its slope in steps until it reaches the lowest point of that function.”

It is a very efficient yet straightforward approach to fit linear regressors and classifiers under convex functions.



3. Results

3.1 Image Classification error

An error rate of **0.304** was attained on the Kaggle project test dataset. The individual performances of the **four** architectures have been illustrated in the table below.

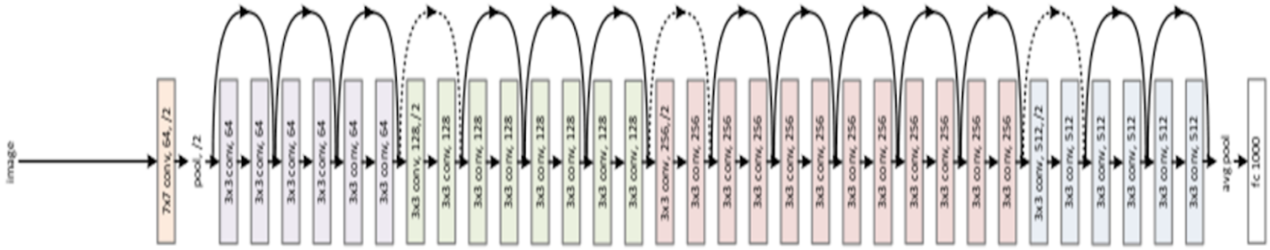
Architecture	**Train Accuracy**	**Validation Accuracy**	**Test Accuracy**
Scratch Model	**0.1637**	**0.1857**	**0.11457**
ResNet 18 (No frozen layers)	**0.7334**	**0.7040**	**0.69537**
Xception	**0.56589**	**0.53200**	**0.48**
VGG16	**0.43300**	**0.37400**	**0.29428**

We achieved our best results (accuracy) for **ResNet 18.**

ResNet 18:

ResNet-18 (stands for Residual learning network) is a Convolution Neural Network {CNN} which in its depth is 18 layers deep. As the network goes deeper and deeper, it becomes more challenging to train. Initially, it is pre-trained and can classify images such as a pencil, book, etc. The layers need to be explicitly reformulated while learning residual functions regarding the layer inputs, instead of learning unreferenced functions.

The depth of representations is fundamental for this task. Solely due to our intense descriptions, we obtain a 20% relative improvement on the Kaggle image detection dataset. It allows several architectural configurations, allowing us to achieve a suitable ratio between the speed of work and quality.



4. Future Applications

The field of computer vision is a remarkably progressive area for research. Computer vision, as the name suggests, is the functionality of artificially intelligent machines to “observe” the world around it just like all animals and make thoughtful analyses according to the given situation, just like humans. It has always been a thread of expanding interest and meticulous research in computer science for years.

It defines a foundation in reproducing human vision. The research in this arena would yield us with a functional machine that can recognize images and visuals, making decisions accordingly. Although, the process of evaluating images is very complex and different problems would require branched out research hovering over all the obstacles that we will have to overcome.

Some of the current ongoing applications of computer vision include :

1. **Healthcare:**
   They were diagnosing diseases by analyzing images obtained by CT scans and other medical imaging processes. Machines can recognize elements and objects from diagnosis with the same accuracy as to how a human doctor does, and although the diagnosis can be made through medical textbook knowledge, a machine can be trained by making it go through specific data and in some cases can also identify patterns by going through the patient's medical records which might get missed by a health professional.

2. **Security:**
   Using biometric analysis such as retinal and fingerprint scanning to identify individuals for security purposes uniquely. One typical example of image recognition is facial recognition in devices used to unlock the screen / make payments. Profound learning algorithms are used in high-security residential complexes and also in businesses in order to prevent corporate espionage.

3. **Manufacturing:**
   Inspecting manufacturing processes and finished products for non-conformance and defects. As we are moving towards fully automated manufacturing processes, we will need to make systems more intelligent in order to supervise industry processes and outcomes. Computer vision can act as an aid to technologies like the Internet of Things {IOT}, improving the autonomous processes. E.g., defects and non - conformities in manufactured products can be caught using Computer Vision, eliminating the need for an inspection by humans in the assembly line.

4. **Transportation:**
   Guiding autonomous vehicles by identifying obstacles, people, and road signs along the way. Self-driving cars are a concept of the futuristic world, where cars are able to detect the barriers using the cameras installed on it that can scan the surrounding periphery around the vehicle so that it automatically orients itself according to its surrounding. It can also read road signs and follow all the traffic rules for the user's maximum safety.

5. References

- **https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf**
- **https://www.pyimagesearch.com/2016/08/01/lenet-convolutional-neural-network-in-python/**
- **https://keras.io/guides/transfer_learning/**

- **https://www.deeplearningbook.org/**

- **https://arxiv.org/pdf/1206.5533.pdf**
- **https://cs231n.github.io/optimization-2/**
- Word Vectors I- https://www.youtube.com/watch?v=e_IJNzOaQY4
- Word Vectors II - https://youtu.be/yDIdShhExSk
- https://www.youtube.com/watch?v=oH0Q3Bj9EUM
- https://www.youtube.com/watch?v=oH0Q3Bj9EUM
- https://www.youtube.com/watch?v=ZFYjy-yBWmg
- https://www.youtube.com/watch?v=8rXD5-xhemo&list=PLoROMvodv4rOhcuXMZkNm7j3fVwBBY42z&index=1
- https://www.youtube.com/watch?v=kEMJRjEdNzM&list=PLoROMvodv4rOhcuXMZkNm7j3fVwBBY42z&index=2