Automated image classification for heritage photographs using Transfer Learning of Computer Vision in Artificial Intelligence

Viratkumar K. Kothari[1] and Dr Sanjay M. Shah[2]

[1]Ph.D. Scholar, Kadi Sarva Vishwavidyalaya, Gandhinagar, Gujarat
[2]Director, Narsinhbhai Institute of Computer Studies & Management, Kadi, Gujarat

Article History: Received: 10 June 2019; Revised: 12 July 2019; Accepted: 27 August 2019; Published online: 10 September 2019

Abstract:
There is substantial archival data available in different forms, including manuscripts, printed papers, photographs, videos, audio, artefacts, sculptures, buildings, and others. Media content like photographs, audio, and videos are crucial content because such content conveys information well. The digital version of such media data is essential as it can be shared easily, available on the online or offline platform, easy to copy, easy to transport, easy to back up, and easy to keep multiple copies in different places. The limitation of the digital version of media data is the lack of searchability, as it hardly has any text that can be processed for OCR. These important data cannot be analysed and, therefore, cannot be used in a meaningful way. To make this data meaningful, one has to manually identify people in the images and tag them to create metadata. Most of the photographs were possible to search based on very basic metadata. This data when hosted on the web platform, searching media data is becoming a challenge due to its data formats. Improvement in existing search functionality is required to improve the searchability of the photographs in terms of ease of usage, quick retrieval and efficiency. The recent revolution in machine learning, deep learning, and artificial intelligence offers a variety of facilities to process media data and identify meaningful information from it. This research paper explains the methods used to process digital photographs to classify people in the given photographs, tag them, and save that information in the metadata. We will tune various hyperparameters to improve their accuracy. Machine learning, deep learning, and artificial intelligence offer several benefits, including auto-identification of people, auto-tagging them, providing insights, and finally, the most important part is that it drastically improves the searchability of photographs.

It was envisaged that about 85% of the manual tagging activity might be reduced, and the searchability of photographs would be improved by 90%.

Keywords – Deep Learning, Transfer Learning, Convolutional Neural Networks, Image Classification, Image Processing, Machine Learning, Computer Vision

I. INTRODUCTION

Photographs are significantly important as they convey information visually, even after many years. It is an essential part of the Heritage. The photographs exactly provide information on a series of events that happened at a particular point in time. It helps to understand a piece of time very accurately. Therefore, the photographs are crucial evidence of the events that happened in the past.

We will use older photographs to experiment with the classification of people in the photographs. In earlier days, the size of the photographs was not standardised. So, the sizes of the photographs taken with different cameras were different. It sometimes differs from the model of the camera. One of the major challenges with old photographs is there are very few people available who can classify a specific person from all the images. It becomes even more difficult when those images are of different sizes, black and white and old.

The physical photographs, once converted into digital format, offer numerous benefits. But, it provides only limited searchability by default. Most of its searchability is based on metadata. So, if metadata can be improved, a lot of searchability will also be improved. Apart from that, it also helps to interlink various photographs based on people, places, events, etc. Technologies like machine learning, deep learning and artificial intelligence may help here.

We will use digitised content on Mohandas Gandhi for this purpose. Gandhiji played an important role in the Indian independence movement. He has visited about 2,500 places in India and abroad. Most of these locations have become heritage sites now. There is enormous physical content available at various places. This content includes letters, books, manuscripts, photographs, audio, videos, artefacts, and buildings. Media content, such as photographs, audio, and videos, is in analogue format. Many of this content has already been digitised along with its metadata. The metadata provides information and insight about the content, e.g. the metadata for photos, maybe

size of the photograph, type of photograph (colour or black & white), photographer, date of the photo, people in the photo, and place of the photograph, etc.

The Gandhi Heritage Portal is an online platform where digitised content and metadata are hosted. It can be accessed using the link www.gandhiheritageportal.org. It is one of the largest authentic repositories on life, thought and works of Mahatma Gandhi. The existing search on photographs works only on metadata. The metadata contains general information about the photographs, such as the size of the photographs, photographer, event, resolution, and, in some cases, the names of a few people in the photographs. The reason behind not having the names of all the people in the photographs is it is generally difficult to identify people because images are of low resolution, images are black and white and lack of domain experts who can identify those people. Therefore, the photographs have basic metadata, but the classification of a person in most of the photographs is still to be done. However, some of the persons can be identified from the photographs, which then can be used as base data for further automated image processing. The automated process of classifying people from the photographs can be done using machine learning.

The auto-classification of people in images will not only help in classifying people, but that data may also be used as core metadata. This metadata can be used to interlink the photograph with other related data, e.g., if we have identified a person from the photographs, an automated procedure can then interlink the photograph with books, journals, videos, places and many other digital data available on the portal.

This paper is organised as follows:

Section I Introduction, Section II Related Work, Section III Information about the digital Data, image classification architecture and approach IV Environmental Setup, Section V Results and Observations of the digital platform, Section VI Conclusion and future work.

II. RELATED WORK

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton [1] have introduced the AlexNet architecture, a deep convolutional neural network (CNN), which achieved a significant breakthrough in image classification accuracy on the ImageNet dataset. By employing multiple layers of convolutional and pooling operations with rectified linear unit (ReLU) activations, AlexNet demonstrated the power of deep learning in computer vision tasks. Its success paved the way for subsequent advancements in deep learning research and applications. Karen Simonyan and Andrew Zisserman [2] proposed the VGG network architecture, characterised by a deep stack of convolutional layers with small 3x3 filters. Despite its simplicity, VGG achieved state-of-the-art performance on image classification tasks, showcasing the importance of depth in convolutional neural networks. Its modular architecture and straightforward design principles have made it a popular choice for various computer vision applications.

Szegedy et al. [3] introduced the Inception-v3 architecture, which employed deep convolutional networks with multiple branches of varying kernel sizes to capture features at different scales. This approach enabled the model to achieve improved accuracy on image classification tasks while maintaining computational efficiency. Kaiming He, Xiangyu Zhang, Jian Sun, and Shaoqing Ren [4] have proposed a deep residual learning framework that addressed the challenges of training very deep neural networks by utilising shortcut connections. ResNet demonstrated unprecedented depths of up to 152 layers and achieved state-of-the-art performance on image classification benchmarks, showcasing the effectiveness of residual learning in overcoming optimisation difficulties. Their groundbreaking work laid the foundation for training extremely deep networks and revolutionised the field of deep learning by introducing a novel approach to network optimisation and training. Howard et al. [5] introduced MobileNets, a family of efficient convolutional neural networks designed for mobile and embedded vision applications. By employing depth-wise separable convolutions and parameterisation techniques, MobileNets achieved state-of-the-art accuracy with significantly fewer parameters, making them suitable for resource-constrained devices. Their efficient architectures paved the way for deploying deep learning models on mobile devices and edge computing platforms, enabling a wide range of real-world applications.

Chollet [6] proposed the Xception, a deep learning architecture that replaced traditional convolutional layers with depthwise separable convolutions. By decoupling spatial and cross-channel correlations, Xception achieved improved efficiency and performance on image classification tasks, outperforming previous CNN architectures. His comprehensive experimental evaluations and theoretical analyses provided insights into the benefits of depthwise separable convolutions and their applications in various deep-learning tasks. Hu et al. [7] introduced Squeeze-and-Excitation (SE) Networks, which incorporated channel-wise attention mechanisms to adaptively recalibrate feature maps. By emphasising informative features and suppressing irrelevant ones, SE Networks achieved improved performance on image classification tasks, surpassing previous state-of-the-art architectures.

Zagoruyko and Komodakis [8] proposed Wide Residual Networks (WRNs), which extended the ResNet architecture by widening the network and increasing the number of channels. WRNs achieved superior performance on image classification tasks by leveraging wider networks to capture richer feature representations and reduce overfitting. Their extensive empirical evaluations and ablation studies highlighted the advantages of wide networks in improving model capacity and generalisation ability, contributing to advancements in deep learning architectures.

Xie et al. [9] introduced ResNeXt, a scalable architecture that aggregated multiple paths within residual blocks to increase model capacity. By leveraging a cardinality parameter to diversify feature representation, ResNeXt achieved state-of-the-art performance on image classification benchmarks with improved efficiency and accuracy. Huang et al. [10] proposed DenseNet, a densely connected convolutional network architecture that maximised feature reuse and facilitated gradient flow. By densely connecting each layer to every other layer in a feed-forward fashion, DenseNet achieved state-of-the-art performance on image classification tasks with improved parameter efficiency.

Wang et al. [11] introduced the Residual Attention Network (RAN), a novel architecture that integrated residual learning with attention mechanisms for image classification. By selectively attending to informative regions within an image, RAN achieved improved discriminative capability and enhanced feature representation compared to traditional CNNs. Their extensive experiments and ablation studies demonstrated the effectiveness of attention mechanisms in enhancing network performance and interpretability, paving the way for further exploration of attention-based models in computer vision tasks. Zhang et al. [12] proposed a deep convolutional neural network (CNN) architecture for image denoising, leveraging residual learning to learn the residual between noisy and clean images directly. The proposed network achieved state-of-the-art denoising performance by training on large-scale datasets, surpassing traditional methods based on handcrafted features.

Byeongho Heo et al. [13] introduced knowledge distillation, a technique for transferring knowledge from a large teacher network to a smaller student network by matching the activation boundaries formed by hidden neurons. By distilling knowledge from a powerful teacher model, the student model achieved comparable performance to the teacher model while being more computationally efficient. Sermanet et al. [14] introduced OverFeat, a unified framework for image recognition, localisation, and detection based on convolutional neural networks (CNNs). By jointly optimising for recognition, localisation, and detection tasks, OverFeat achieved state-of-the-art performance on various computer vision benchmarks, demonstrating the effectiveness of end-to-end learning.

The research papers presented showcase the evolution of deep learning architectures for image classification and object recognition, culminating in highly accurate and efficient models like ResNet, DenseNet, and Squeeze-and-Excitation Networks. Future research possibilities include exploring domain adaptation techniques, zero-shot learning methods, incremental learning, and transfer learning algorithms to improve model generalisation and adaptability.

III. INFORMATION ABOUT THE DIGITAL DATA, IMAGE CLASSIFICATION ARCHITECTURE AND APPROACH

A. Digital data that needs to be processed

The amount of data in the digital form is increasing like never before. This digital data is available in a structured, semi-structured and unstructured form. About eighty per cent of the total data available in the digital form is in an unstructured format. It is always easy to search and interlink the structured and semi-structured data because they are available in table format containing rows and columns. On the other side, unstructured data is always difficult to search because of its format. The data may be in the form of images, videos, audio and free-flow texts. Recent technologies like Machine Learning (ML) and Artificial Intelligence (AI) help to process these types of data and perform a meaningful search on them, e.g., Natural Language Processing (NLP) may be used to process free-flow texts and find meaningful summaries or perform intelligent searches. Computer Vision (CV), a sub-field of Artificial Intelligence, may help to process images, videos and audio.

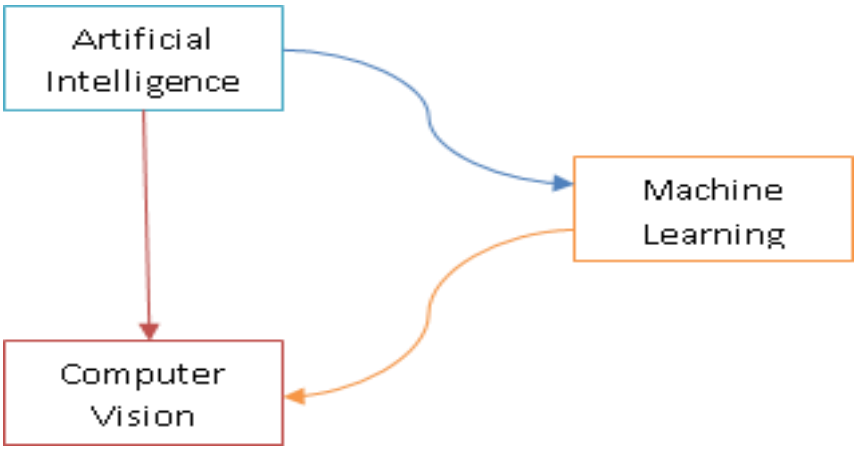

Figure 1: Relationship between Artificial Intelligence and Machine Learning

The images that we want to process contain various types of images, including photographs, images of stamps, posters or the pages of the book. To make the images more meaningful, we should add metadata to them. Such metadata is generally prepared by humans. Such metadata may include fields like the size of the photo, resolution, dimension, name of photographer, place where a photo is taken and the people in the photograph.

It is envisaged that some of the metadata may be auto-generated using technology, e.g., size, resolution, and dimension can be identified, and one of the important tasks of classifying photos by people in it may be achieved using Computer Vision, often abbreviated as CV.

Computer Vision is a field of study that helps a computer to "see" and understand photographs, videos, etc. It is a scientific and interdisciplinary field that deals with how computers can see and understand digital images and videos. It seems simple but quite complex as we still do not completely understand how biological vision works and processes visual data. It is because of its dynamic perception and infinite variety. Recent digital equipment like digital cameras, mobiles, etc., captures high-resolution images. The computers can accurately detect and measure the difference between the colours. But understanding those images is a problem that computers have been struggling with for a long. A computer only sees them as an array of pixels or numerical values.

Computer vision performs a series of activities to process image or video data, including acquiring, processing, analysing, and finally extracting features in the form of numeric data. This numeric data will help a computer understand the digital data and, if needed, can help to compare it with other digital data to find differences, similarities, matches, and patterns. This will help the machine to learn and understand images and videos.

B. Image Classification

Image classification is one of the most important and fascinating tasks that can be performed using computer vision. It allows images to be classified into a set of pre-defined categories. The classification of images into two categories is called binary classification, e.g., classification of images in images of dogs and cats. On the other hand, the classification of images in more than two categories is called multi-class classification, e.g., classifying images into categories like a flower, mountain, sun, dog, cat, bus, scooter, computer, gun and temple. Here, the classification is in 10 categories, so it is a 10-class classification.

There are various types of tasks that computer vision can perform:

1) *Image classification*, where a computer will classify images into two or more pre-defined categories.
2) *Localisation*, where the objective is not only to classify an image but where that object is within the image, e.g., classify an image as a dog image but draw a border in the image where the dog is in the image.
3) *Object detection*, is when a computer will identify how many different objects are in the image, e.g. if there is a cat, dog, and scooter in a single image. The computer will only draw a box around the object but will not be able to identify the object.
4) *Object identification*, is when a computer not only identifies different objects but also names them. So, basically, it will draw a box around an object like a dog and label it as a dog.
5) *Instance segmentation*, is when a computer identifies an object in the image and draws an exact border

around the object rather than a box around it.

6) The scope of this research paper is to classify images into two or more categories.

C. Transfer Learning and Model Selection

There are various techniques for image classification, including Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Multi-Layer Perceptron (MLP), Convolutional Neural Networks (CNN) etc. Also, a hybrid approach is used for image classification. There are pre-trained models such as Alexnet, Googlenet, VGG16, VGG19 Resnet, and many more are available. Such models are accurately trained for multi-class classification. We may import a suitable model and modify it according to our needs, which may give more accurate results than a completely new model. Such an approach is called the Transfer Learning approach and gives much better and more accurate results.

The Deep Convolutional Neural Network models generally take a very long time to train on very large datasets. Reusing the weights of the pre-trained models may significantly save training time. Such models are developed for standard benchmarked computer vision datasets, such as the ImageNet or VGG16, and image recognition tasks. Such best-performing models can be directly used and integrated into other newly developed models for various computer vision problems. The process of using a pre-trained model in a newly developed model is called Transfer Learning.

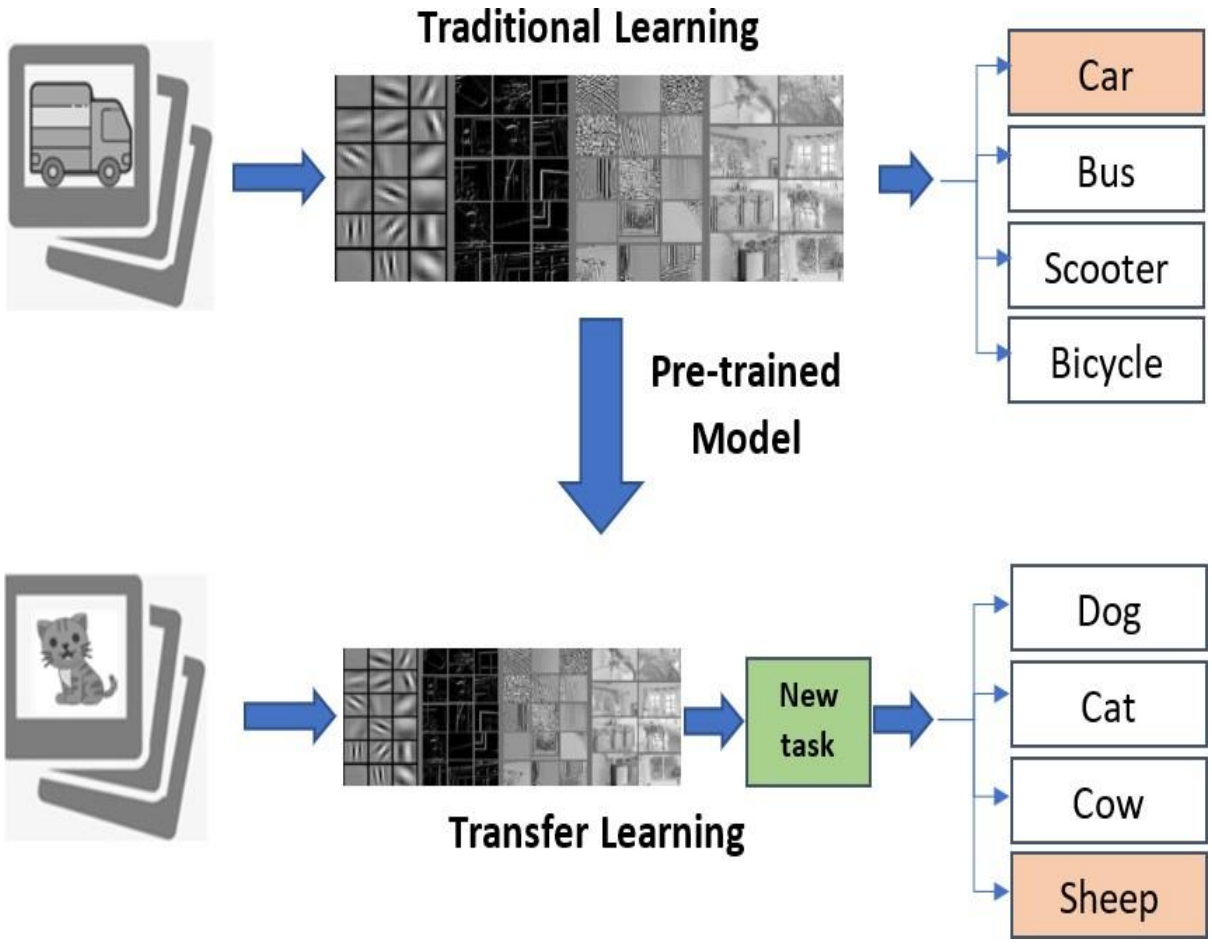

Figure 2: Transfer learning

Transfer Learning is one of the best techniques in machine learning and is widely used for various tasks, including image classification. The VGG is a convolutional neural network with a specific architecture for large-scale image classification. The VGG has two separate architectures: VGG16, which contains 16 layers and VGG19, containing 19 layers. Both architectures are equally good, but we have used VGG16 for image classification. This contains the different parts, including convolution, pooling and fully connected layers. The architecture starts

with two convolution layers and one pooling layer in the first block. The following image depicts the architecture of the VGG16:

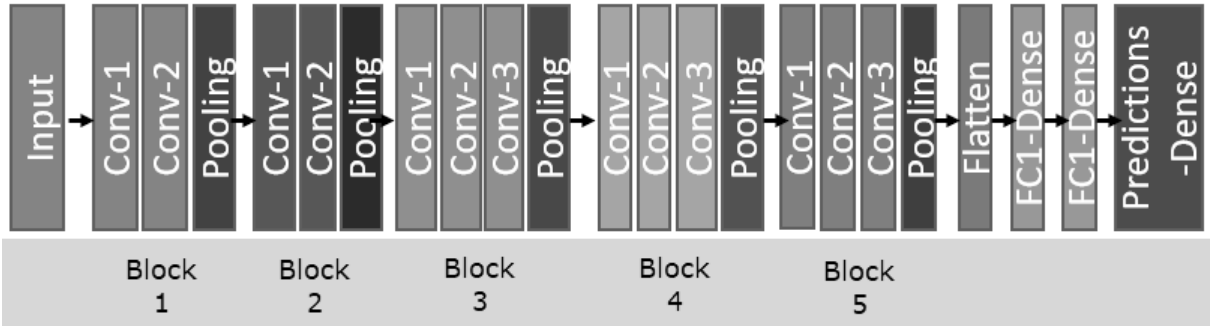

Figure 1: Architecture of VGG16

```
Summary of default VGG16 model.

Model: "vgg16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
fc1 (Dense)                  (None, 4096)              102764544
_________________________________________________________________
fc2 (Dense)                  (None, 4096)              16781312
_________________________________________________________________
predictions (Dense)          (None, 1000)              4097000
=================================================================
Total params: 138,357,544
Trainable params: 138,357,544
Non-trainable params: 0
```

Figure 2: Details of layers in VGG16

There are five blocks in total, and each block has a combination of convolution and pooling.

The model starts with the Input layer. The first and second blocks contain two convolutions and one max-pooling layer, while the third, fourth and fifth layer contains three convolutions and one max-pooling layer. A flatten layer is introduced after block five. Finally, two fully connected layers are introduced just before the prediction. The last layer is the prediction layer.

The Transfer Learning using VGG16 offers various benefits:

1) Learning ability: The model is trained with one lakh images for one thousand categories, and that's why the model can easily detect generic features. Such models have a very high ability to learn.
2) Performance: The models are trained with a high number of images for multi-categories and fine-tuned at their best for the highest level of accuracy. So, reusing such a model also improves performance.
3) Easy availability: The model weights are provided in the form of downloadable files, or in some cases, they provide a convenient API to use the model. This way, the models can be integrated into the new model easily.

Transfer Learning, in simple terms, refers to a process where a model trained for one problem is used for another related problem. Transfer learning, a method in deep learning, involves leveraging knowledge from a pre-trained model to enhance the training of another model tackling a related task. This saves huge infrastructure and training time. The weights of some layers are reused as a starting point for the training, and necessary changes are made for the new problem.

D. Properties of the model

As explained above, the VGG16 model has 16 layers in total. It starts with the input layer, five-block, each containing some convolution layer and max-pooling layer. In the end, there is a flatten layer, two fully connected layers and a final prediction layer for the number categories in which images need to be classified.

The first layer – The input layer takes images as input for the model. The entry layer takes images of the size 224 x 224, and as it accepts colour images, the third parameter is 3. So, this input layer accepts colour images of the size 224 x 224 x 3. Following the input layer, the images will pass through several convolution layers and max-pooling layers.

After block five, a flatten layer is added. The flatten layer removes all the dimensions except for one from the data. It reshapes the tensor to have an equal number of elements contained in it. Flattening can be understood as making a one-dimensional array of elements. The flattening is required to pass the data to the dense layer.

The dense layer is 4096 units, which will stop negative values from being forwarded through the network. A 1000-unit dense layer, in the end, has SoftMax activation. The 1000 units here are several classes with the images that need to be categorised. This means the image will be classified in one of the 1000 categories to which it belongs.

E. Customisation in the layers of the model

The default VGG16 model cannot be used straightforwardly for the custom problem of image classification.

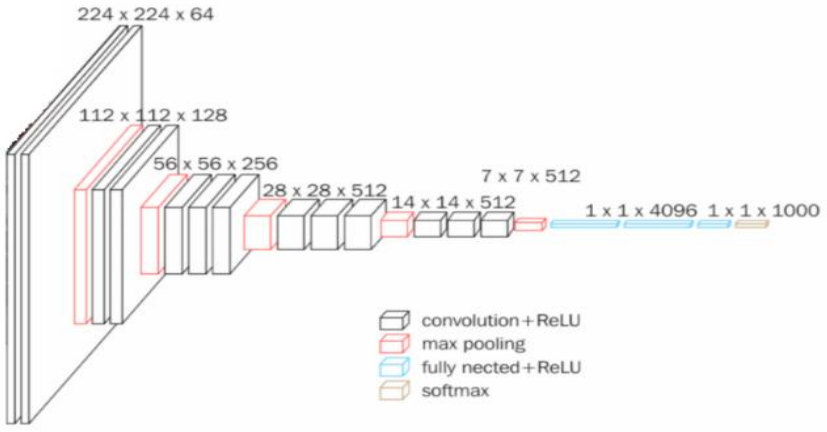

Figure 5: Default VGG16 architecture
[Image Source: https://neurohive.io/en/popular-networks/vgg16/]

Few customisations will be required to fit it for our image classification. We need to classify images into ten different categories. Therefore, we will do the following customisation in layers to fit the default VGG16 model to our problem of 10-class image classification:

1) We will carry the weights of the original VGG16 model to our model
2) The default input layer accepts colour images of the size 224 x 224. So, we need to resize our images to the size 224 x 224. We will change the first layer accordingly
3) The last layer specifies the number of categories in which we need to classify the images. The default is 1000. Our images need to be classified into ten categories. So, we need to change the last (top) layer from 1000 to 10 categories.
4) We will not touch any layers except the first (bottom) and last (top) layers. So, all the layers except the first (bottom) and last layer (top) are made non-trainable.

5) Parameter tuning of the model

The following parameters will be tuned in order to get better performance from VGG16:

1) weights='imagenet': this is to use weights from a pre-trained model
2) input_tensor=input_layer: this is to add a custom input layer as below:
   input_layer=layers.Input(shape=(224,224,3))
3) include_top=False: this is to add a custom top layer in order to classify images into ten categories. It will remove the following layers from the default VGG16 model:

   block5_pool (MaxPooling2D) (None, 7, 7, 512) 0

   flatten (Flatten) (None, 25088) 0

   fc1 (Dense) (None, 4096) 102764544

   fc2 (Dense) (None, 4096) 16781312

   predictions (Dense) (None, 1000) 4097000
4) We will extend the neural network by adding the following layers:

   Adding flatten layer:

   flatten=layers.Flatten()(last_layer)
5) The following set of custom layers may be added to improve the performance:
   dense1=layers.Dense(100,activation='softmax')([last_layer])

   One or more copies of the above layers may be added. We will measure performance with a different set of layers.
6) Adding output (last) layer with ten, i.e., number of classification categories along with SoftMax activation:

   output_layer=layers.Dense(10,activation='softmax')(last_layer)

IV. ENVIRONMENTAL SETUP

We will use the following to test the model:

1) Jupyter Notebook in Kaggle environment (default)
2) RAM: 16 GB
3) GPU: NVIDIA K80 GPU (default)
4) TensorFlow: 2.1 or above

5) Image dataset: The dataset was developed by Mario on Kaggle and has a CC0 licence in the public domain. This contains images of monkeys. It contains images of 10 species. This consists of two folders, viz., training and validation. Each folder contains ten sub-folders containing images of monkeys. Following is the detail of the folder name and corresponding train and validation images:

Label,	Latin Name	Common Name	Train Images	Validation Images
n0	alouatta_palliata	mantled_howler	131	26
n1	erythrocebus_patas	patas_monkey	139	28
n2	cacajao_calvus	bald_uakari	137	27
n3	macaca_fuscata	japanese_macaque	152	30
n4	cebuella_pygmea	pygmy_marmoset	131	26
n5	cebus_capucinus	white_headed_capuchin	141	28
n6	mico_argentatus	silvery_marmoset	132	26
n7	saimiri_sciureus	common_squirrel_monkey	142	28
n8	aotus_nigriceps	black_headed_night_monkey	133	27
n9	trachypithecus_johnii	nilgiri_langur	132	26

Figure 6: Information about the dataset used for the experiment

The total number of images is about 1,400 of the 400 x 300 or larger resolution.

The reason for using this dataset is it is intended to test the fine-grain classification tasks, and it can be best used with Transfer Learning. The dataset may be accessed using the following URL:

https://www.kaggle.com/slothkong/10-monkey-species

V. RESULTS AND OBSERVATIONS

In this section, the observations are presented based on tests:

1) Combination – 1:

Sr. No.	Processor	No. of added layer	Dense Value	Dropout Value	Epochs	Batch Size	Final Loss	Accuracy	Validation Loss	Validation Accuracy	Interpretation
1	GPU	1 D + 1 DO + 4 D + 1 DO	100	0.3	20	128	0.12	0.12	2.52	0.16	Poor accuracy

We have added the following combination of layers, but it gives very poor accuracy.

dense0=layers.Dense(100,activation='relu')(flatten) dense1=Dropout(0.3)(dense0)
dense2=layers.Dense(100,activation='relu')(dense1) dense3=layers.Dense(100,activation='relu')(dense2)
dense4=layers.Dense(100,activation='relu')(dense3) dense5=layers.Dense(100,activation='relu')(dense4)
dense6=Dropout(0.3)(dense5)

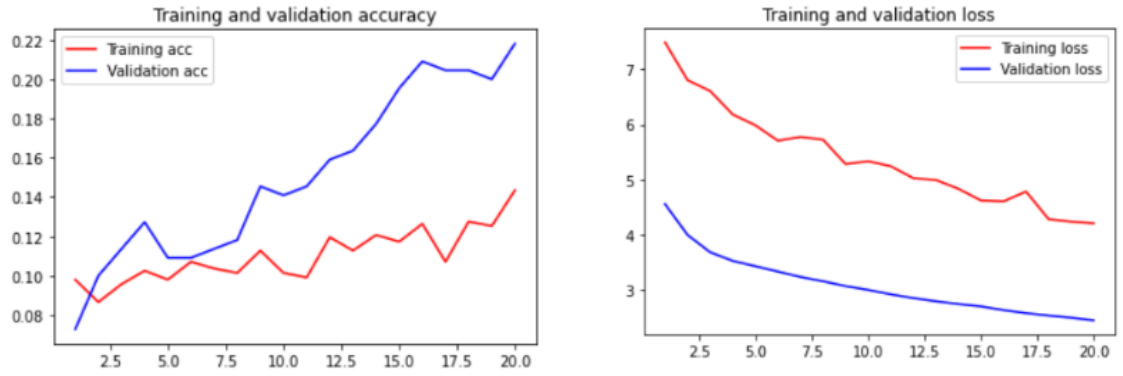

Figure 7: Training Vs Validation Accuracy & Loss (Combination -1)

2) Combination – 2:

Sr. No.	Processor	No. of added layer	Dense Value	Dropout Value	Epochs	Batch Size	Final Loss	Accuracy	Validation Loss	Validation Accuracy	Interpretation
2	GPU	1 D + 1 DO + 3 D	100	0.3	20	128	3.77	0.16	2.75	0.22	Poor accuracy

We have added the following combination of layers, but it also gives very poor accuracy.

dense0=layers.Dense(100,activation='relu')(flatten) dense1=Dropout(0.3)(dense0)
dense2=layers.Dense(100,activation='relu')(dense1) dense3=layers.Dense(100,activation='relu')(dense2)
dense4=layers.Dense(100,activation='relu')(dense3)

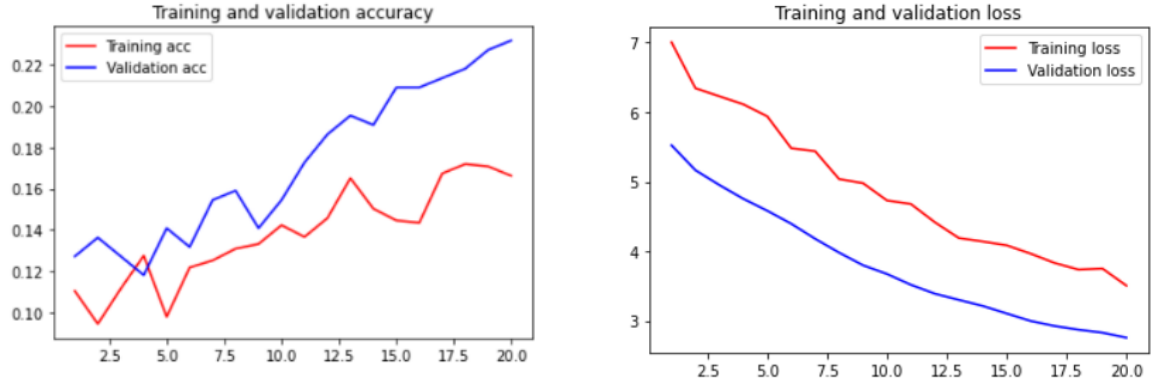

Figure 8: Training Vs Validation Accuracy & Loss (Combination -2)

3) Combination – 3:

Sr. No.	Processor	No. of added layer	Dense Value	Dropout Value	Epochs	Batch Size	Final Loss	Accuracy	Validation Loss	Validation Accuracy	Interpretation
3	GPU	1 D + 1 DO + 1 D	100	0.3	20	128	8.05	0.21	5.81	0.25	Poor accuracy

We have added the following combination of layers, but it also gives very poor accuracy.

dense0=layers.Dense(100,activation='relu')(flatten) dense1=Dropout(0.3)(dense0)
dense2=layers.Dense(100,activation='relu')(dense1)

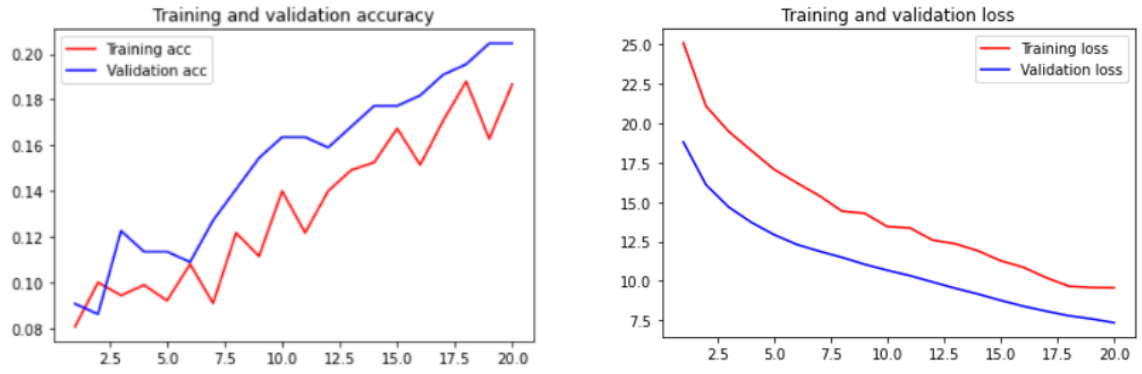

Figure 9: Training Vs Validation Accuracy & Loss (Combination -3)

4) Combination – 4:

Sr. No.	Processor	No. of added layer	Dense Value	Dropout Value	Epochs	Batch Size	Final Loss	Accuracy	Validation Loss	Validation Accuracy	Interpretation
4	GPU	1 D + 1 DO	100	0.3	20	128	11.57	0.29	9.03	0.30	Poor accuracy

We have added the following combination of layers, but it also gives very poor accuracy.

dense0=layers.Dense(100,activation='relu')(flatten)

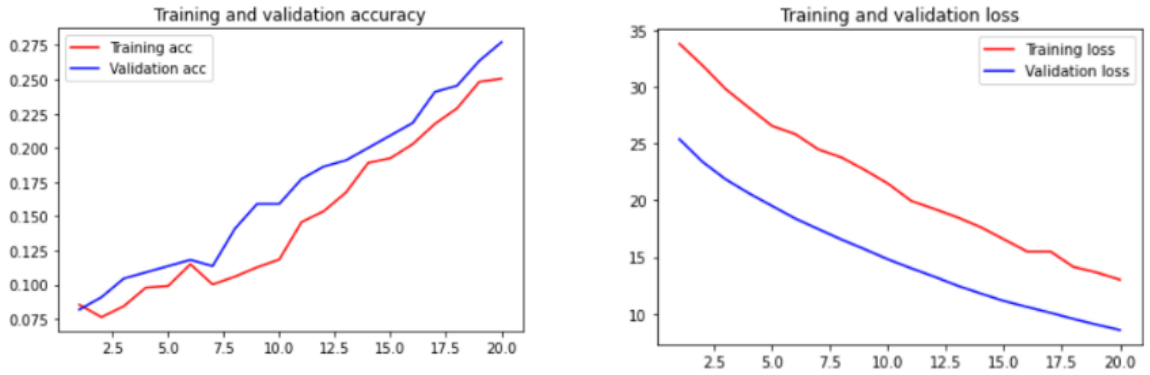

Figure 10: Training Vs Validation Accuracy & Loss (Combination -4)

5) Combination – 5:

Sr. No.	Processor	No. of added layer	Dense Value	Dropout Value	Epochs	Batch Size	Final Loss	Accuracy	Validation Loss	Validation Accuracy	Interpretation
5	GPU	5 D + 1 DO	10K,9 K,8K,7 K,6K	0.85	20	128	2.54	0.69	0.73	0.84	Good accuracy but underfit

We have added the following combination of layers, which gave good accuracy, but the model is underfitting.

dense1=layers.Dense(100,activation='relu')(flatten)

dense2=layers.Dense(100,activation='relu')(dense1)

dense3=layers.Dense(100,activation='relu')(dense2)

dense4=layers.Dense(100,activation='relu')(dense3)

dense5=layers.Dense(100,activation='relu')(dense4)

dense6=Dropout(0.3)(dense5)

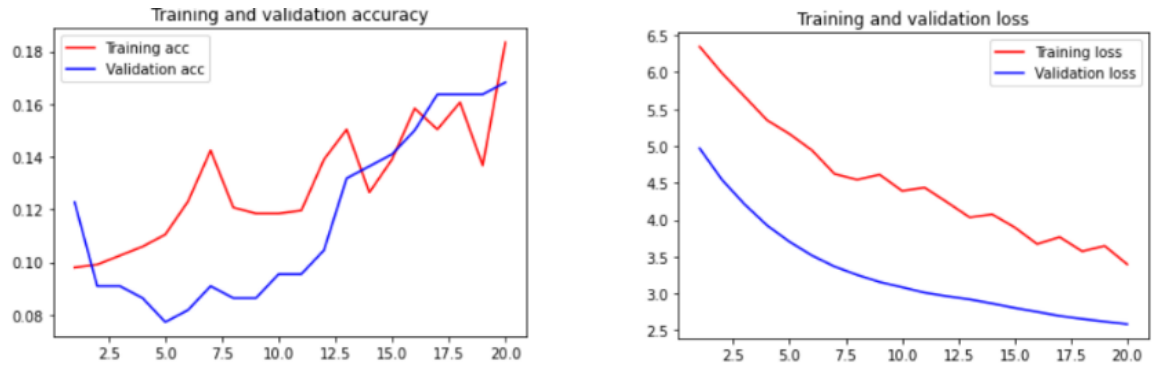

Figure 11: Training Vs Validation Accuracy & Loss (Combination -5)

6) Combination – 6:

Sr. No.	Processor	No. of added layer	Dense Value	Dropout Value	Epochs	Batch Size	Final Loss	Accuracy	Validation Loss	Validation Accuracy	Interpretation
6	GPU	0 D + 0 DO	NA	NA	20	128	2.75	1.00	1.37	0.95	Best accuracy but very little overfit

We have applied the default VGG16 model with no added layers, and that gives the **best accuracy and has very little overfit**.

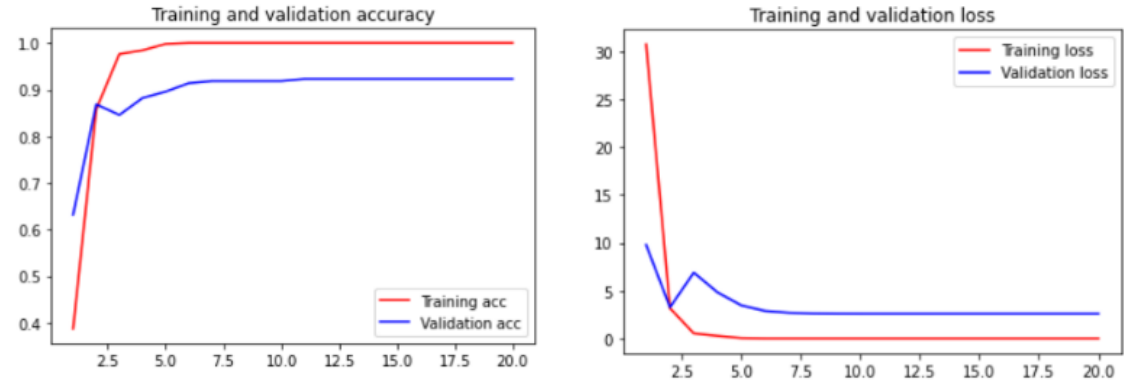

Figure 12: Training Vs Validation Accuracy & Loss (Combination -6)

VI. CONCLUSIONS AND FUTURE WORK

Following is the summary of the experiment with various combinations of layers added to the model. The best accuracy with very little overfit is available by applying the default VGG16 model without adding any custom layers

.

Sr. No.	Processor	No. of added layer	Dense Value	Dropout Value	Epochs	Batch Size	Final Loss	Accuracy	Validation Loss	Validation Accuracy	Interpretation
1	GPU	1 D + 1 DO + 4 D + 1 DO	100	0.3	20	128	0.12	0.12	2.52	0.16	Poor accuracy
2	GPU	1 D + 1 DO + 3 D	100	0.3	20	128	3.77	0.16	2.75	0.22	Poor accuracy
3	GPU	1 D + 1 DO + 1 D	100	0.3	20	128	8.05	0.21	5.81	0.25	Poor accuracy
4	GPU	1 D + 1 DO	100	0.3	20	128	11.57	0.29	9.03	0.30	Poor accuracy
5	GPU	5 D + 1 DO	10K,9K,8K,7K,6K	0.85	20	128	2.54	0.69	0.73	0.84	Good accuracy but underfit
6	GPU	0 D + 0 DO	NA	NA	20	128	2.75	1.00	1.37	0.95	Best accuracy but very little overfit

Figure 13: Summary of the combination of the layers

The Transfer Learning model with a custom Input layer and Output (top) layer gives the best accuracy with minimal overfit. Thus, this provides a quite accurate classification of the images with an accuracy of 95 per cent and may be applied to the digital heritage data for image classification.

REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton., "Imagenet classification with deep convolutional neural networks", Journal of Neural Information Processing Systems (NeurIPS), 2012

[2] Karen Simonyan, Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", Proceedings of the International Conference on Learning Representations (ICLR), 2014

[3] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, "Going Deeper with Convolutions", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun., "Residual Networks of Residual Networks: Multilevel Residual Networks", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016

[5] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", arXiv preprint arXiv:1704.04861, 2017

[6] François Chollet., "Xception: Deep Learning with Depthwise Separable Convolutions", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

[7] Jie Hu, Li Shen, Gang Sun., "Squeeze-and-Excitation Networks", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018

[8] Sergey Zagoruyko, Nikos Komodakis., "Wide Residual Networks", Proceedings of the British Machine Vision Conference (BMVC), 2016

[9] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He, "ResNeXt: Aggregated Residual Transformations for Deep Neural Networks", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

[10] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger, "Densely Connected Convolutional Networks", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

[11] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, Xiaoou Tang, "Residual Attention Network for Image Classification", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

[12] Kai Zhang, Wangmeng Zuo, Lei Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising", IEEE Transactions on Image Processing, 2017

[13] Byeongho Heo, Minsik Lee, Sangdoo Yun, Jin Young Choi, "Knowledge Transfer via Distillation of Activation Boundaries Formed by Hidden Neurons", The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19), 2018

[14] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, Yann LeCun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks", International Conference on Learning Representations (ICLR), 2014.