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

Viratkumar K. Kothari¹ and Dr Sanjay M. Shah²

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

Abstract:
There is substantial archival data available in different forms, including manuscripts, printed papers, photographs, videos, audios, artefacts, sculptures, buildings, and others. Media content like photographs, audios, and videos are crucial because such content conveys information well. The digital version of such media data is essential as it can be shared easily, available in the online or offline platform, easy to copy, easy to transport, easy to back up and easy to keep multiple copies at 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 out of it. This research paper explains the methods to process digital photographs to classify people in the given photographs, tag them and saves that information in the metadata. We will tune various hyperparameter to improve their accuracy. Machine learning, deep learning and artificial intelligence offers several benefits, including auto-identification of people, auto-tagging them, provide insights and finally, the most important part is it improves the searchability of photographs drastically.

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

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

I. INTRODUCTION
The photographs are significantly important as it covey 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 size of photographs taken from different cameras was 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 difficult when those images are of different size, black and white and old.

The physical photographs, once converted into digital format, offers 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, place, event etc. The technologies like machine learning, deep learning and artificial intelligence may help here.

We will use digitised content on Mohandas Gandhi for this purpose. He performed an important role in the Indian independence movement. He visited about 2,500 places in his life in India and abroad. Most of such locations have become heritage site now. There is enormous physical content available at various places. This content includes letters, books, manuscripts, photographs, audios, videos, artefacts, and buildings. The media content like photographs, audios and videos are in analogue format. Many of this content is already 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 is hosted along with its metadata. 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 like the size of the photographs, photographer, event, resolution and in some cases name of few people in the photographs. The reason behind not having the name 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. Although, 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 then can 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 architecture and development specification IV Environmental Setup, parameter tuning and testing scenarios, Section V Results and Observations of the digital platform, Section VI Conclusion and future work.

II. RELATED WORK

Konstantinos Papadopoulos, Anis Kacem, Abdelrahman Shabayek, and Djamila Aouada [1] has explained that face identification/recognition techniques has improved a lot in the last few years. They explained that the most common approach to do this to rely on static RGB picture frames and on its neutral expressions of the face. This method, however, ignores important facial shape cues and also facial deformations because of expressions that can lead to an impact on performance. They proposed a new framework for identification/recognition for dynamic 3D face based on certain facial key points. The sequence of the facial expressions is represented as a Spatio-temporal graph. This graph is constructed using some three-dimensional facial landmarks. Each node of the graph contains texture and local shape features extracted from its neighbourhood. They have used a Spatio-temporal Graph Convolutional Network (ST-GCN) for face identification/recognition.

Rinku Datta Rakshit, et al. [2] have a demonstration that how low resolution and very low-resolution image of the face captured by surveillance cameras can be used for face recognition. They have explained that if low and high-resolution images together used as test data to learn face identification, the performance of face identification will be degraded. They have proposed a Cross-Resolution face identification system to address this issue. They are using a deep convolutional neural network with different types of pooling operations, which extracts robust resolution features from high, low and very low-resolution images.

Lacey Best-Rowden et al. [3] explains that the new challenges are encountered when face recognition applications progress from constrained sensing with cooperative subjects scenarios to unconstrained scenarios with uncooperative subjects. An example may be a driving licence or video surveillance. They further explain that this is due to ambient light variation. They have used various sources of information about the person, including video tracks, images, 3D models and a sketch generated based on verbal descriptions provided by others. This method gives improved results as it uses various sources for learning compared to a single source in the case of traditional technology. Himanshu S. Bhatt et al. [4] discusses about-face recognition algorithms which are generally trained on high-resolution images and which performs better on high-resolution images. The performance of such systems degrades when using low-resolution images as test data. They have used transfer machine learning to enhance the performance of cross-resolution face recognition. The experiments were done on multiple face databases to test the efficacy. They have also demonstrated the usefulness of the proposed approach for tough images databases.

Rinku Datta et al. [5], on the other side, talks about a face identification system that works with the face images that are visible, look-alike and post-surgery using a novel approach that exploits from local graph structure (LGS). This attempt improves the performance of the face identification system also under the influence of changes in pose, expression, illumination, make-up and accessories. Andrea F. Abate et al. [6] talks about the investment of a huge amount of resources to improve security systems by using biometric data, including face recognition, fingerprints or iris scans. Biometrics is a good alternative but suffers from various drawbacks. Fingerprints are socially accepted against the Iris scan, which is reliable but intrusive. On the other hand, any biometrics require consent from users. Their framework allows users to browse and filter based on some pre-defined categories and improve the results of matching biometrics. Shaymaa M. Hamandi, et al. [7] explains that the extraction robust facial features are an effective and important process for face recognition and identification system. These should remain unchanged to scale, illumination, and rotation etc. They also explain that there are myriad feature extraction techniques available to improve the accuracy of face identification and recognition. Bhawna Ahuja et al. [8] discusses Local Binary Pattern (LBP) based Extreme Machine Learning. This helps to identify high-dimensional face images of different resolutions. They have used Local Binary Pattern to represent micro-regions from the image of the faces in the form of feature vectors. These are then concatenated as face descriptors for the representation of face images. Guodong Guo et al. [9] explains the usage of the Support Vector Machine (SVM)
for face recognition. The SVM is a recent technique and is used for pattern recognition. The authors have explained the usage of SVM along with binary tree recognition to tackle the face recognition or identification problem. They have used the Cambridge ORL face database, which consists of about four hundred images of 40 individuals. The images contain different types of poses, expressions, and facial details. They have also used another larger database containing more than a thousand images of one hundred and thirty-seven people. They have compared the results of both facial databases.

Yunyan Wang, Chongyang Wang, Lengkun Luo, and Zhigang Zhou [10] have used a quite new approach to Transfer Learning. Transfer Learning is based on Convolution Neural Network (CNN). They explained that HOG features of the training sample which are similar to the test data, is extracted, and then an SVM classifier is used. At last, the pre-classification results are used as training samples to train the Transfer Network of CNN to build a new transfer model. This model can be used to classify the test or the final images. Their experiment shows that the accuracy of the classification is improved quite a lot, and overall classification accuracy is improved up to 95%. While comparing it with the traditional classification model, the accuracy is improved by 5% with this method. [11] Annegreet Van Opbroek, Hakim C. Achterberg et al. [11] proposes using Transfer Learning for image segmentation through combining weights of images and kernel learning. They have mainly explored medical images. They explain that the medical image segmentation methods are mostly based on supervised classification, which generally performs well. However, problems may arise when the training dataset and test dataset follows distributions, e.g., different cameras or scanners, scanning protocols and patient groups. Under such circumstances, the accuracy of the overall result impacted. They have proposed to use kernel learning as a way to decrease differences between training and test datasets. Finally, they proposed combining image weight and kernel learning for improvement in performance. Emine Cengil et al. [12] mentions that deep learning technologies have been successfully used in several fields for years, and image classification is one of such fields. They suggest using the Transfer Learning approach, which pre-trained models such as Alexnet, Googlenet, VGG16, VGG19 Resnet, and many more. This may be easily used for image classification. The results show an acceptable performance rate, and VGG16/19 performs best for image classification.

Euijoon Ahn, Ashnil Kumar et al. [13] has discussed the accuracy and robustness of image classification with supervised deep learning, which requires a large amount of annotated data. The annotation is normally to be done manually due to its complexity. They suggest that Transfer Learning may be used here to overcome this problem by using a generic feature extractor which should be trained with large-scale general images and fine-tune generic knowledge with a smaller number of annotated images. Their approach has higher image classification accuracy than the transfer learned approach and competitive with supervised fine-tuned methods. Manali Shaha, Meenakshi Pawar [14] mentions that the Convolutional Neural Network (CNN) is capable of doing robust feature extraction and information mining. It has been used for image classification, object recognition, image super-relation etc., due to its supreme feature extraction capabilities. Among many pre-trained models, VGG16 and VGG19 perform better for most image classification. They have used GHIM10K, and CalTech256 databases for their experiments, and their analysis shows that the fine-tuned VGG19 architecture outperformed the other CNN and hybrid learning approach for image classification.

III. INFORMATION ABOUT THE DIGITAL DATA, IMAGE CLASSIFICATION ARCHITECTURE AND APPROACH
A. Digital data that needs to be processed
The 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 make 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, audios 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 out meaningful summary or perform intelligent searches. Computer Vision (CV), a sub-field of Artificial Intelligence, may help to process images, videos and audios.
The images that we want to process contain various type 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, 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 is quite complex as we have still not completely understood how biological vision works and processes the visual data. It is because of its dynamic perception and infinite variety. The 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 since along. A computer only sees them as an array of pixels or numerical values.

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

B. Image Classification

Image classification is one of the important and fascinating tasks performed using Computer Vision. It allows classifying images in a set of pre-defined categories. The classification of images in two categories is called binary classification, e.g., classification of images in images of Dogs and images of Cats. On the other hand, the classification of images in more than two categories is called multi-class classification, e.g., classify 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**, where a computer will identify how many different objects are there in the image, e.g., 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**, where a computer will not only identify different objects but also name them. So, basically, it will draw a box around an object like a dog and label it as a dog.

5) **Instance segmentation**, where a computer will identify an object in the image and draw an exact border around the object rather than drawing a box around it.

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), and Convolutional Neural Networks (CNN) etc. Also, there is a hybrid approach 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 suitable model and modify according to our need may give more accurate results than a completely new model. Such an approach is called the Transfer Learning approach and gives much better and accurate results.

The Deep Convolutional Neural Network models generally take a very long time to train on very large datasets. Re-using 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 etc., 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 into a newly developed model is called Transfer Learning.

Transfer Learning is one of the best techniques in machine learning 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 architecture: VGG16, which contains 16 layers and VGG19, containing 19 layers. Both the architectures are equally good, but we have used VGG16 for image classification. This contains the different part, including convolution, pooling and fully connected layers. The architecture starts with two convolution layer and one pooling layer in the first block. The following image depicts the architecture of the VGG16:

Figure 2: Transfer learning
Figure 3: Architecture of VGG16

Summary of default VGG16 model.

<table>
<thead>
<tr>
<th>Layer Type</th>
<th>Output Shape</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>input_2 (InputLayer)</td>
<td>[None, 224, 224, 3]</td>
<td>0</td>
</tr>
<tr>
<td>block1_conv1 (Conv2D)</td>
<td>(None, 224, 224, 64)</td>
<td>179</td>
</tr>
<tr>
<td>block1_conv2 (Conv2D)</td>
<td>(None, 224, 224, 64)</td>
<td>36928</td>
</tr>
<tr>
<td>block1_pool (MaxPooling2D)</td>
<td>(None, 112, 112, 64)</td>
<td>0</td>
</tr>
<tr>
<td>block2_conv1 (Conv2D)</td>
<td>(None, 112, 112, 128)</td>
<td>73856</td>
</tr>
<tr>
<td>block2_conv2 (Conv2D)</td>
<td>(None, 112, 112, 128)</td>
<td>147584</td>
</tr>
<tr>
<td>block2_pool (MaxPooling2D)</td>
<td>(None, 56, 56, 128)</td>
<td>0</td>
</tr>
<tr>
<td>block3_conv1 (Conv2D)</td>
<td>(None, 56, 56, 256)</td>
<td>295168</td>
</tr>
<tr>
<td>block3_conv2 (Conv2D)</td>
<td>(None, 56, 56, 256)</td>
<td>590080</td>
</tr>
<tr>
<td>block3_conv3 (Conv2D)</td>
<td>(None, 56, 56, 256)</td>
<td>590080</td>
</tr>
<tr>
<td>block3_pool (MaxPooling2D)</td>
<td>(None, 28, 28, 256)</td>
<td>0</td>
</tr>
<tr>
<td>block4_conv1 (Conv2D)</td>
<td>(None, 28, 28, 512)</td>
<td>1180160</td>
</tr>
<tr>
<td>block4_conv2 (Conv2D)</td>
<td>(None, 28, 28, 512)</td>
<td>2359808</td>
</tr>
<tr>
<td>block4_conv3 (Conv2D)</td>
<td>(None, 28, 28, 512)</td>
<td>2359808</td>
</tr>
<tr>
<td>block4_pool (MaxPooling2D)</td>
<td>(None, 14, 14, 512)</td>
<td>0</td>
</tr>
<tr>
<td>block5_conv1 (Conv2D)</td>
<td>(None, 14, 14, 512)</td>
<td>2359808</td>
</tr>
<tr>
<td>block5_conv2 (Conv2D)</td>
<td>(None, 14, 14, 512)</td>
<td>2359808</td>
</tr>
<tr>
<td>block5_conv3 (Conv2D)</td>
<td>(None, 14, 14, 512)</td>
<td>2359808</td>
</tr>
<tr>
<td>block5_pool (MaxPooling2D)</td>
<td>(None, 7, 7, 512)</td>
<td>0</td>
</tr>
<tr>
<td>flatten (Flatten)</td>
<td>(None, 25088)</td>
<td>0</td>
</tr>
<tr>
<td>fc1 (Dense)</td>
<td>(None, 4096)</td>
<td>102764544</td>
</tr>
<tr>
<td>fc2 (Dense)</td>
<td>(None, 4096)</td>
<td>16781312</td>
</tr>
<tr>
<td>predictions (Dense)</td>
<td>(None, 1000)</td>
<td>4097000</td>
</tr>
</tbody>
</table>

Total params: 138,357,544
Trainable params: 138,357,544
Non-trainable params: 0

Figure 4: 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 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 image for one thousand categories, and that’s why the model can easily detect generic features. Such models have a very high ability of learning.

2) Performance: The models are training with a high number of images for multi-categories and fine-tuned at their best for the highest level of accuracy. So, re-using 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 is a deep learning technique where a pre-trained model is used to train another model for a similar problem. 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 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 an input to 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 that has an equal number of elements contained in the tensor. 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 of 4096 units which will stop forwarding negative values 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 straight forward for the custom problem of image classification.

Figure 5: Default VGG16 architecture

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 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) layer. So, all the layers except the first (bottom) and last layer (top) are made non-trainable.

5) **Parameter tuning of the model**

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

1) weights=’imagenet’: this is to use weights from pre-trained model

2) input_tensor=input_layer: this is to add custom input layer as following:
   
   ```
   input_layer=layers.Input(shape=(224,224,3))
   ```

3) include_top=False: this is to add custom top layer in order to classify images in 10 categories. It will remove 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 neural network by adding 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 copy 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 is developed by Mario on Kaggle, having 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 of that folder contains ten sub-folders containing images of monkeys. Following is the detail of folder name and corresponding train and validation images:

![Image](image_url)

**Figure 6: Information about 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:

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Processor</th>
<th>No. of added layer</th>
<th>Dense Value</th>
<th>Dropout Value</th>
<th>Epochs</th>
<th>Batch Size</th>
<th>Final Loss</th>
<th>Accuracy</th>
<th>Validation Loss</th>
<th>Validation Accuracy</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GPU</td>
<td>1 D + 1 DO + 4 D + 1 DO</td>
<td>100</td>
<td>0.3</td>
<td>20</td>
<td>128</td>
<td>0.12</td>
<td>0.12</td>
<td>2.52</td>
<td>0.16</td>
<td>Poor accuracy</td>
</tr>
</tbody>
</table>

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

```python
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)
```
2) **Combination – 2:**

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Processor</th>
<th>No. of added layer</th>
<th>Dense Value</th>
<th>Dropout Value</th>
<th>Epochs</th>
<th>Batch Size</th>
<th>Final Loss</th>
<th>Accuracy</th>
<th>Validation Loss</th>
<th>Validation Accuracy</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>GPU</td>
<td>1 D + 1 DO + 3 D</td>
<td>100</td>
<td>0.3</td>
<td>20</td>
<td>128</td>
<td>3.77</td>
<td>0.16</td>
<td>2.75</td>
<td>0.22</td>
<td>Poor accuracy</td>
</tr>
</tbody>
</table>

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

```python
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 7: Training Vs Validation Accuracy and Loss (Combination -1)](image)

3) **Combination – 3:**

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Processor</th>
<th>No. of added layer</th>
<th>Dense Value</th>
<th>Dropout Value</th>
<th>Epochs</th>
<th>Batch Size</th>
<th>Final Loss</th>
<th>Accuracy</th>
<th>Validation Loss</th>
<th>Validation Accuracy</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>GPU</td>
<td>1 D + 1 DO + 1 D</td>
<td>100</td>
<td>0.3</td>
<td>20</td>
<td>128</td>
<td>8.05</td>
<td>0.21</td>
<td>5.81</td>
<td>0.25</td>
<td>Poor accuracy</td>
</tr>
</tbody>
</table>

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

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

![Figure 8: Training Vs Validation Accuracy and Loss (Combination -2)](image)
We have added a combination of layers but it also gives very poor accuracy.

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

We have added following combination of layers which gave good accuracy but model is underfit.

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 and Loss (Combination -5)

6) Combination – 6:

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

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 and Loss (Combination -6)

VI. CONCLUSIONS AND FUTURE WORK

Following is the summary of the experiment with various combination 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.
The Transfer Learning model with 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 the accuracy of 95 per cent and may be applied to the digital heritage data for image classification.

REFERENCES


