RVL-CDIP : Transfer learning for multi-class image classification

Sanidhya Singh **P. Madhav**

1. IIT Kanpur and sanidhyasingh80@gmail.com
2. IIT Kanpur and madhav3921@gmail.com

Abstract

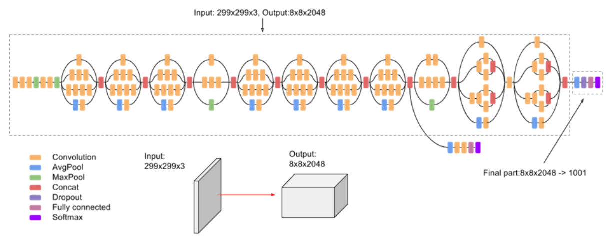
In recent years machine learning is playing a vital role in our everyday life. It can be used for weather forecasts, routing unmanned aerial vehicles and self-driving cars, recommendation systems in various social media platforms, automating employee access control, etc. Computer vision and image processing are excelling in the field of segmentation, feature extraction, and object detection from image data. The advancement of artificial neural networks and the development of deep learning architectures such as the convolutional neural network(CNN), which is based on artificial neural networks has triggered the application of multiclass image classification and recognition of objects belonging to multiple categories.
Here we have trained various transfer learning models from Keras using deep convolutional neural networks and analyzed their performances on our data set, containing 16000 images with 1000 images belonging to each class. The dataset was collected from the RVL-CDIP dataset. And finally presented the performance of InceptionV3 which performed better than some of the other models on our dataset.

Keywords: Deep learning, transfer learning, Convolutional Neural Network(CNN), Object detection, Multi-class Image classification, Keras

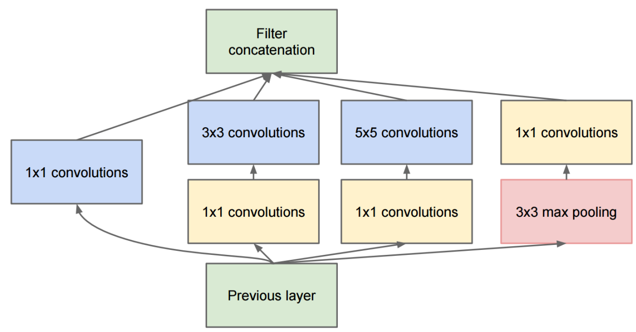
Feature Selection/Method Description

As per the summarized results of 3 epochs of different models shown below, we have tested our datasets on 4 models, namely MobileNet, ResNet50, VGG16, and InceptionV3, with sufficient epoch size. Some models like NASNetLarge also have performed well giving one of the highest accuracies of all models, but due to our computational power insufficiency, we were unable to proceed with such models for further training.
Based on the validation accuracies we got initially, different types of analysis as mentioned in novelty, and the computational resources we have, we have selected InceptionV3 as our final model, since it's also computationally less expensive.
Inception v3 is a widely-used image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset and around 93.9% accuracy in the top 5 results.

Model Architecture:



Model's Naive form:



Experimental Results

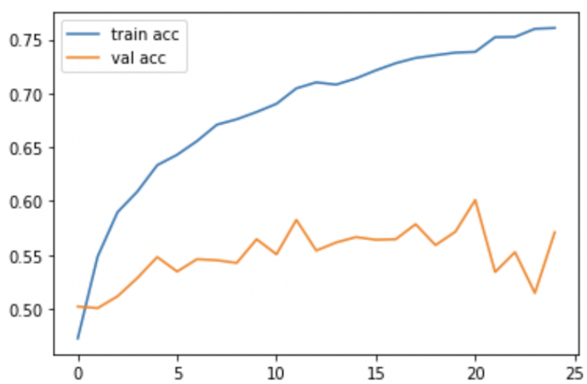
We ran all the pre-trained models with a small epoch size of 3 from Keras library to find the best fit for our dataset, we obtained following results:

Model name	Model Params	Validation accuracy
MobileNetV3Small	1529968	0.15831664
MobileNetV2	2257984	0.43887776
MobileNet	3228864	0.54308617
MobileNet	3228864	0.53907818
MobileNetV3Large	4226432	0.16232465
NASNetMobile	4269716	0.45891786
DenseNet121	7037504	0.47695392
DenseNet169	12642880	0.49498999
VGG16	14714688	0.38276553
DenseNet201	18321984	0.51903808
VGG19	20024384	0.37474951
Xception	20861480	0.46492988
InceptionV3	21802784	0.51703405
ResNet50V2	23564800	0.53306615
ResNet50	23587712	0.27655312

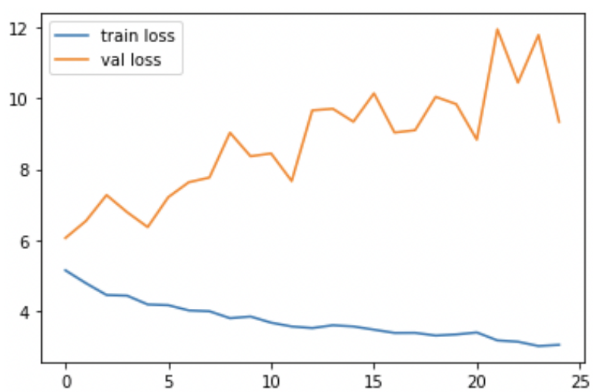
ResNet101V2	42626560	0.45490983
ResNet101	42658176	0.24048096
InceptionResNetV2	54336736	0.49098197
ResNet152V2	58331648	0.49098197
ResNet152	58370944	0.20240481
NASNetLarge	84916818	0.44288579

We tested our datasets on 4 models, namely MobileNet, ResNet50, VGG16 and InceptionV3, with sufficient epoch size. We came to find that InceptionV3 is giving best results among the aforementioned models. We then increased the epoch size to 25 to obtain an accuracy of 76.07% on train dataset and 57.1% on validation dataset.

Figures and Tables



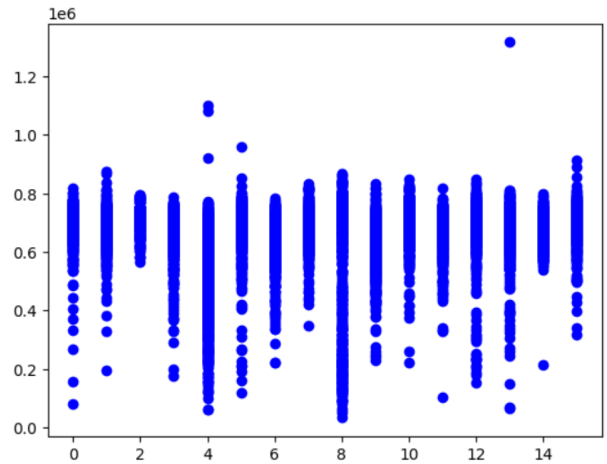
Training accuracy and Validation accuracy vs epochs for InceptionV3 model on given dataset



Training loss and Validation loss vs epochs for InceptionV3 model on given dataset

Novelty

From the given datasets and labels we first created separate directories for train and validation, in which we created 16 folders, one for each class. After studying the model's architecture, to get a better fit for such a huge model, we doubled the size of the training dataset by creating duplicates of each image and tuned several parameters such as batch size, etc. accordingly for different models.
After testing all transfer learning models of multi-class image classification from Keras and observing the performance of different models we have chosen InceptionV3 finally, which has also given the highest score in the hackathon.
Since we can't train all the models fully and predict the results and check, we have generated various box plots and dot plots comparing images of different classes to check consistency in the predictions of the models.



For example in the above plot of purely white pixels count vs class, we have checked the distortion of data points compared to the data points of that respective class. The more the distortion, the higher the chances of the inaccuracy of the model. So by performing various similar analyses and different types of plots, we have judged the performance of different models and came up with very few of them to train with the given dataset.
We have also tried to get a completely different range of values for any class, that would help in classifying that particular class of images from the test set. For example, the folder class has really less variance in their pixel intensities (peaks in intensity histogram), we can use this information to filter out a few of the classes beforehand to ease the task of our model.

Conclusion

Multi-class image classification needs a lot of preprocessing and deep convolutional neural networks to get the right predictions. Even if some pre-trained model is well known to give higher accuracy, we might not get such high accuracy on our dataset, unless we hypertune the model or preprocess the data. The analysis done in this report is helpful for further training of different models on the actual RVL-CDIP dataset. Based on different reasons as discussed we have trained the model InceptionV3 and generated results based on it and we have found effective results. We still see a lot of things that can be done to further enhance the model efficiency and so we'll continue the research on these and get more fruitful results by building a very efficient and accurate model.

References

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, 2017, *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,* Google Inc.
Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna, 2015, *Rethinking the Inception, Architecture for Computer Vision,* Zbigniew Wojna University College London
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 2015, *Deep Residual Learning for Image Recognition,* Mircorsoft.com
Karen Simonyan, Andrew Zisserman, 2014, *Very Deep Convolutional Networks for Large-Scale Image Recognition,* Department of Engineering Science, University of Oxford

Pretrained models obtained from https://keras.io/api/applications/
Various other resources and websites for application of these models.