Image Classification using Keras with Artificial Neural Networks

Mariam Khalid
khalidm11@coventery.ac.uk
ID: 10076976

Abstract— **The growing amount of available images invites researchers to explore and investigate our world by designing methods to extract useful information. The increased availability of high resolution images allows to sense very detailed structures on the surface of our planet. In this paper, we explore different models for image analysis and formulating models for predicting the places. Detecting such features and attributes will lead to a variety of applications that can help the communities in many ways. To derive insights, We used Intel image classification dataset and applied multiple variants of CNN and well known transfer learning models for predicting different types of natural images. Base CNN model shows accuracy of 70 % and RESNET outperformed from all models with accuracy $>$ 90 %**

I. INTRODUCTION

With advancements in digital technologies in computer vision, different techniques making it possible to explore images precisely and predict tasks of all science fields. Extracting information from high resolution camera, optical remote sensing or satellite images have received much attention recently. Information extracted from photographs has found applications in a wide range of areas including urban planning, crop and forest management, disaster relief, and climate modeling. At present, much of the extraction is still performed by human experts, making the process slow, costly, and error prone The increased availability of high-resolution satellite imagery allows to sense very detailed structures on the surface of our planet. Access to such information opens up new directions in the analysis of remote sensing imagery.

In this paper, we implemented an pre-processing technique motivated from the different methodologies of ETL techniques [1] [2], which obtain data from original source to perform informative analysis and features extraction to extract features against not so rare RGB features with defined ETL process, balancing of our dataset using distributions and validation techniques to predict six different types of natural scenes.

We are following the Intel Image Classification challenge posted by Intel for image classification on Kaggle. The challenge was initially on datahack [1]. The data contains around 25000 images with 14000 for training, 3000 for testing and 7000 for predicting or validation with a the goal is to develop algorithms that can automatically classify natural images based on their respective RGB features. The dataset contains variety of images that were used by experts to classify the images into one of six classes. This is not a trivial task since interpreting images is very challenging even for human specialists. Therefore, modeling the natural scenes will be critical for the success of any approach.

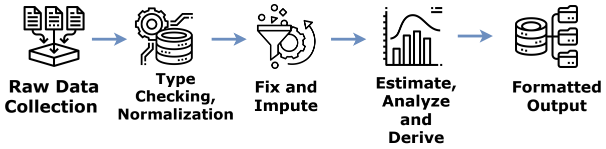

Fig. 1: Pre-processing steps applied on Images .

A predictor that can evaluate scenes based on the images, can not only help in the field of urban analysis, object detection, semantic segmentation, agriculture, exploring space and wide range of medical sciences. One of the major applications also includes the self driving cars and safe cities where live image and video feed can be detected by the application in cars, houses and roads for security, safety and automation of traffic system.

In our effort, we used Kaggle Intel image classification data. Intel image classification is a large, freely-available dataset of images comprising variety of natural scenes containing mountains, forrest, glacier, streets, sea and buildings. After the pre-processing steps as shown in Fig. 1, we used different variant of CNN [3], class imbalance techniques and transfer learning model to achieve the goal of building a neural network which can classify these images with more accuracy, then results of all the models were recorded.

The paper is organized as follow: In section II, we explore the literature review. In section III, we explain the methods and frameworks used. In section IV, we describe our experimental setup and execution of models of V. Further, we discussed the significance of results in section VI and concluded the paper in section VII.

II. Literature Review

Numerous researches have also been conducted on the image data set for creating new possibilities of research and scientific areas. As we are working with a huge data set, the volume, diversity, dimensions and ETL matters. The studies with goal of object detection, computer vision, autonomous systems, semantic segmentation and bio-informatics with both structural and unstructured data have not only help

[1] https://datahack.analyticsvidhya.com

researchers in identifying the new possibilities but also helped them to recognize the places with higher accuracy.

With millions of images being loaded on a daily basis, managing and annotating all the images had previously became a huge problem for developers. Internet using the fast page algorithm already managing and maintaining the indexes but how much can be used for reaserch and development purpose. Deng, Jia, et al. introduced the Imagenet [4], which is a huge database for images containing 15 million images with annotations using the wordnet. Their structure organizes the images in hierarichal order with annotations. They showed that their dataset is more accurate and contains clean high resolution images which helps in research and development in different fields. This dataset has not only help developers to develop algorithm and test it on the 12 subtree annotated dataset which lead to apply such algorithm to other fields and extract features that can be used to solve global problems.

The base model that changed the dimension of computer vision is Convolutional Neural Network. Krizhevsky A, Sutskever I, Hinton GE trained deep convolutional neural network [3] on 1.2 million high resolution imagenet and to optimize they introduced the concept of dropout to reduce the overfitting and created benchmarking results. Krizhevsky, Alex and Hinton, Geoffrey presented their research [5] on the natural images using the RDB (Restricted Boltzmann Machines) [6] for learning the filters and created CIFAR-10.

Training a deep neural network and finding the right features and applying the correct layers is not an easy task. X Glorot, Y Bengio helped us with their study of understanding the difficulties to apply and train the deep neural networks on images [7]. They explained how the initialization of parameters, layers and normalization of affect the network. Moreover, they explained the activation functions roles in optimizing the network and proposed an initialization scheme for better convergence of model. Similarly T Salimans, DP Kingma presented a study [8] of reparameterization of that how weight normalization is affects the network. They used batch-up normalization for speed boost and showed that their model can be utilized in NN and recurrent neural networks. They demonstrated their method to be effective in deep inforsment learning as well.

With the increasing advancements to CNN, K Simonyan, A Zisserman proposed VGG16 – Convolutional Network for Classification and Detection [9]. Their model achieved 92.7% top-5 test accuracy on the already discussed Imagenet dataset. the input of the model for the first convolutional layer was of size 224 x 224 image. Then it is passed to multiple convolutional layers, where they applied filters with a size of 33. In their configurations, they also used 11 convolution filters and applied spatial and max pooling with a stride of 1x1. Fully connected layers have also more depth than the previous models and for non-linearity they use ReLU activation function.

He, Kaiming, et al. presented their model ResNET [10] and handled issues of training time and complexity of VGG16, Their layers were in more depth but showed that the complex is much less due to residual learning framework. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task.

Gradient descent [11] have been evolving with the advancements of models. DP Kingma, J Ba presented Adam [12] which optimizes the stochastic graident technique using the first order gradient and findig the low order moments. We have used Adam for training of our models. CNN have been changing the dimension of computer vision solutions for a different fields. In the study, [13] Szegedy, Christian, et al. benchmarked the ILSVRC 2012 by exploring different ways to scale the network with the aim of getting the computation effectively. They achieved 21.2% top-1 and 5.6% top-5 error.

III. Data

Dataset used is Intel Image Classification challenge data posted by Intel for image classification on Kaggle. The data contains around 25000 images with 14034 for training, 3000 for testing and 7000 for predicting or validation. The dataset contains variety of images that were used by experts to classify the images into one of six classes.

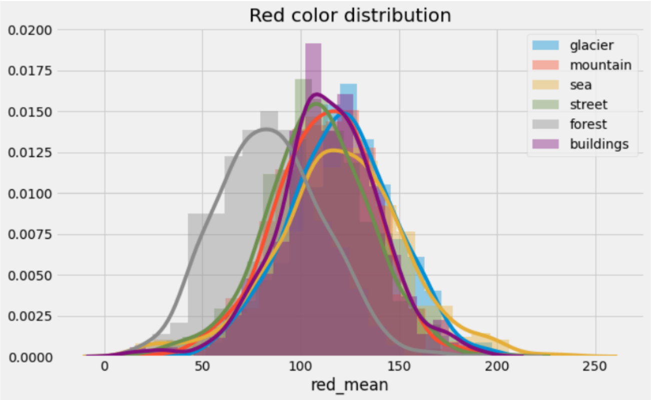

Fig. 2: Red color distribution in training samples

After extraction of dataset, training, testing and prediction images. We applied the preprocessing steps as shown in the Fig 1. The specific steps that were followed are as follow:

1) Reading each image
2) Validate the size of image
3) Resize the image to 150 x 150 x 3
4) Normalizing each image by 255

Dataset	Count
Training	14034
Testing	3000
Prediction	7000

TABLE I: Distribution of Dataset

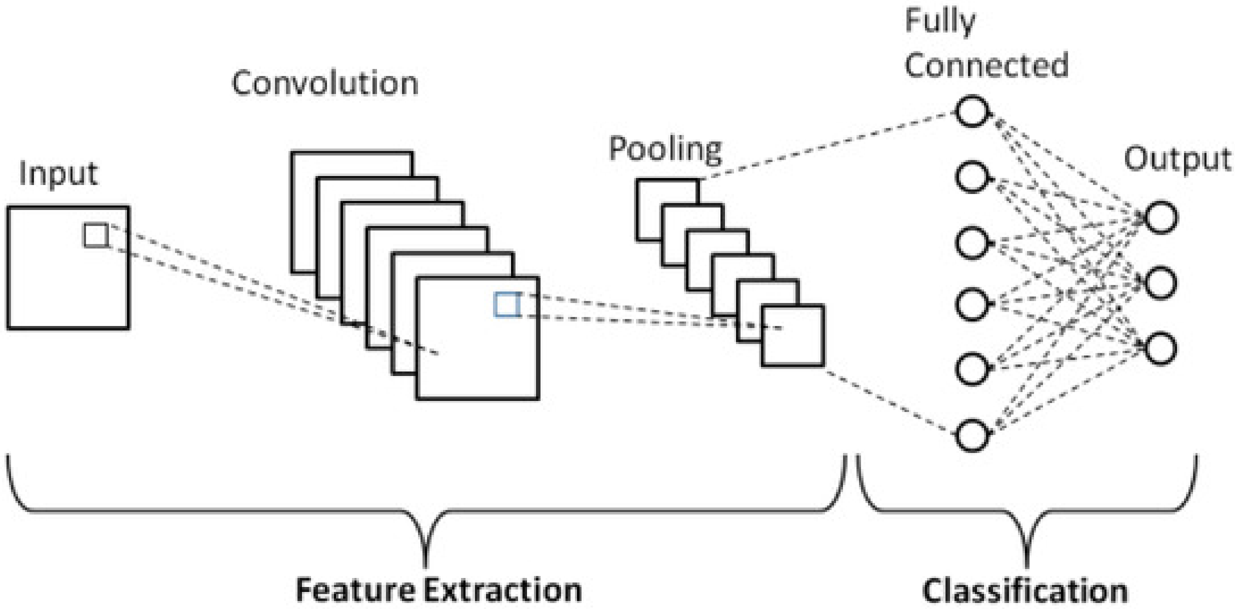

Fig. 3: Convolutional neural network architecture

Explored our training images for how the types of classes is distributed and what is the ratio between them. We also plotted the Red, Green and Blue color distribution. The red color distribution can be seen in Fig. 2

IV. Experimental Setup

We are aiming to classifying images into 6 different classes. We used accuracy as the metric of performance to validate our models which is defined as:

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n} \mathrm{I}(\mathrm{i}_i = \mathrm{i}_i^{'}) \tag{1}$$

V. Models

A. Convolutional Neural Networks

1) Base CNN: Convolutional Neural Network is a deep learning model comprising of convolutional, pooling and connected layers of neurons which assign different learnable parameters to the features of image input pixels. The convolutional layers extract the features which further goes through pooling layers and pass it to fully connected layers.

Class	Count
Building	2191
Forrest	2271
Glacier	2404
Sea	2274
Street	2383
Mountain	2512

TABLE II: Class Count

Generic base convolutional neural netwrok architecture is shown in Fig. 3.

We followed the Base CNN architecture as:

1) 2 Convolutional layers
2) 2 Pooling layers
3) Resize the image to 150 x 150 x 3
4) Normalizing each image by 255

Pooling layers are one of the major component of CNN models. They are used to reduce the space representation of extracted features from convolutional layers as they progress towards output. There are multiple types of pooling. Example of max pooling is shown in Fig. 5.

Activation functions plays an important rule to activate the output for the next progression layer. They handle which neuron to fire against the updated weights to respective targets. We have used three different types of activation functions. Different combinations of activation functions were used at different layers of networks. Sigmoid being one of them which also known as logistic function which transforms the input in range of [0-1]

$$P(x) = \frac{1}{1 + \mathrm{e}^{-x}} \tag{2}$$

The tangent function, or tanh also a nonlinear activation function transforms values between -1.0 and 1.0. tanh function is preferred over sigmoid as models are easier to train using tanh. It is basically the scaled version of sigmoid function.

$$P(x) = tan(x) = \frac{2}{1 + \mathrm{e}^{-2x}} - 1 \tag{3}$$

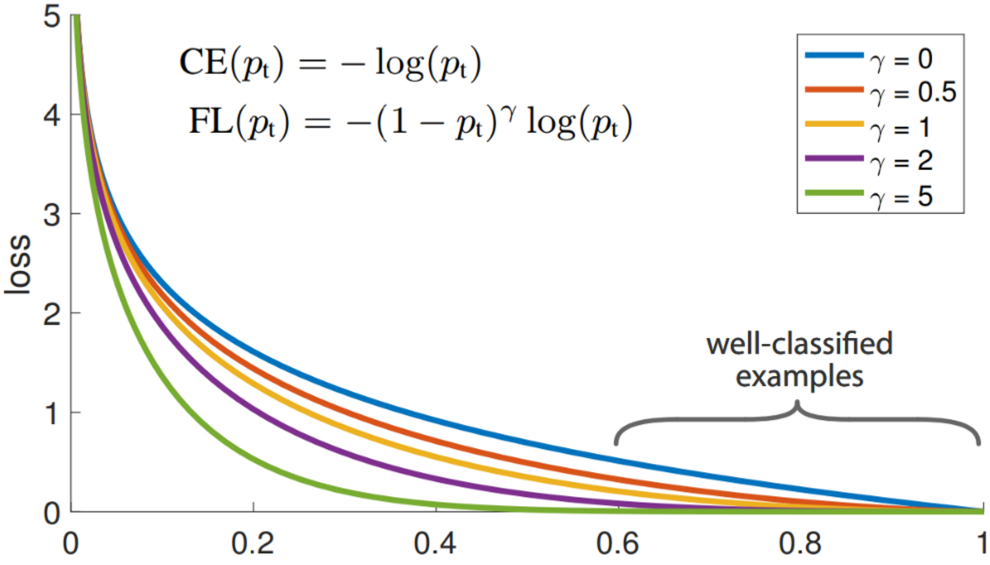

Fig. 4: Increasing the value of γ from 0 reduces the relative loss for well classified examples while placing greater focus on hard misclassified ones

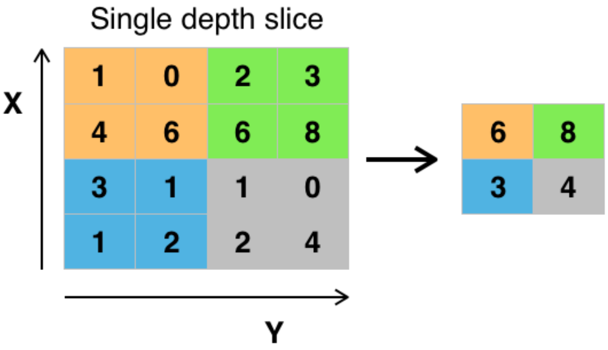

Fig. 5: Pooling example for single step slide

But, to provide more saturation with less complexity and gaining more sensitivity ReLU activation function is used. Applied on layers with rectified function and transforms the input either firing or not firing state.

$$Y = max(0, x) \tag{4}$$

Focal Loss is used to deal with the problem of class imbalance we have used the recently proposed novel loss function called focal loss [14]. Focal loss reduces the relative loss for well classified examples, putting more emphasis on hard misclassified examples. It achieves this by modifying the well known cross entropy loss as follows also shown in Fig. 4:

$$FL = -(1-p_t)\gamma\log(\mathrm{p}_t) \tag{5}$$

B. Transfer Learning

Transfer learning is one of state of the art technique in which trained features are extracted from knowledge stored and applied to different problems to achieve better result and escape the computations. In deep learning tasks the trained knowledge is used for a first step and then adding respective layers to solve one particular problem.

Four different transfer learning models used in this exploration which are as follow:

1) VGG16
2) ResNET
3) DenseNET with Focal Loss
4) InceptionV3

All the models were trained on 20 epochs with different layers as shown in code. Accuracy matric is used for validating our results. For loss function we have used sparse-categorical cross entropy.

$$E_{\text{entropy}} = -\sum_{n}^{N}\sum_{k}^{c} t_k^n \ln y_k^n \tag{6}$$

Platform used for models training and execution is Colab [2] provided by Google. As these bigger models require higher GPUs and TPUs, local machine was unable to train these models. Colab provides GPU computational power and give freedom to test your models. We used their platform for all of our trainings.

[2] colab.research.google.com

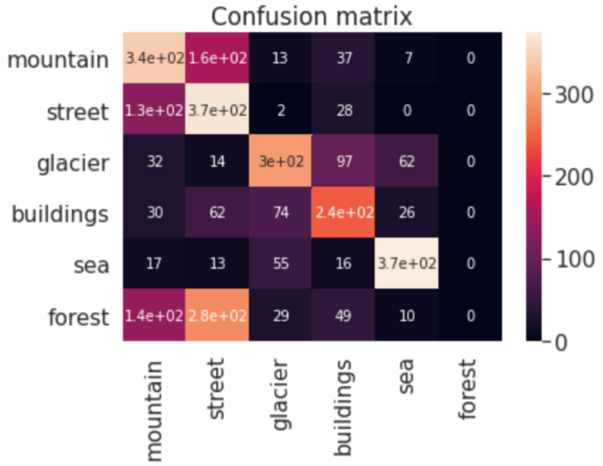

Fig. 6: Confusion matrix against Base CNN

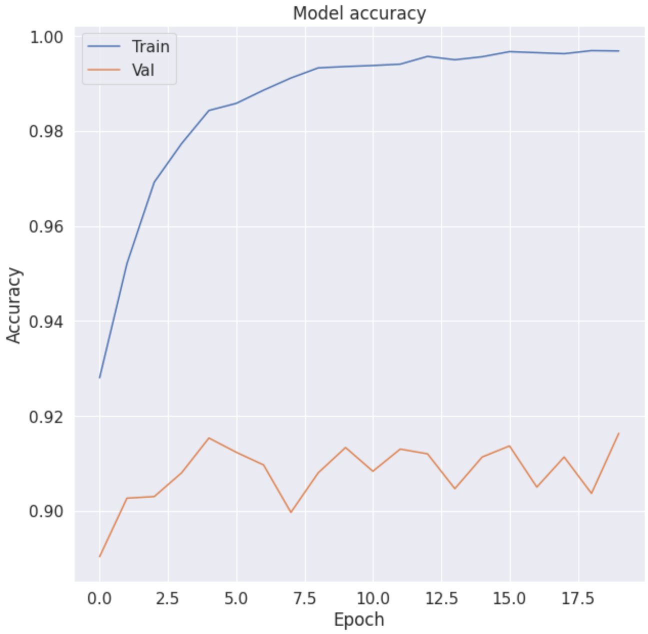

Fig. 7: Training and validation accuracy plot against ResNET Model

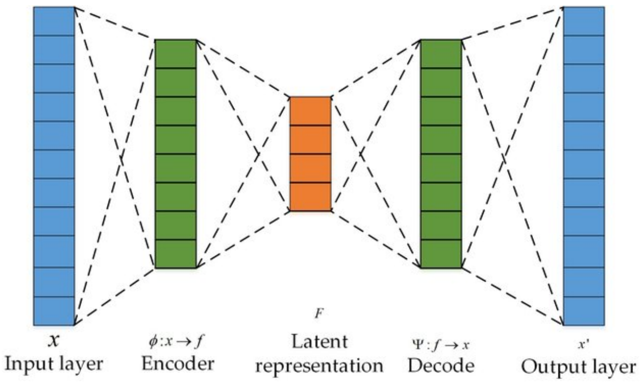

Fig. 8: Encode-Decoder Network Architecture

VI. Results and Discussion

Keeping our goal of classifying multi-class images of natural scenes, we have applied multiple neural network. Our base model was CNN with two convolutional layers and two pooling layers. Followed by another variant of CNN with more convolutional layers and pooling layers. From our variant model CNN, it outrank the base model in accuracy. But, from all trained models transfer learning model outperformed and gave us accuracy of above 90 %. To explore the results we have plot their training and validation plots and shown in the notebook. Furthermore, we have plotted the confusion matrix for CNN models to explore and learn about the different visualizatoin techniqes. Fig. 6. shows the confusion matrix of our base CNN model and Fig 7. shows the model accuracy for training and validation dataset on 20 epochs. Accuracy of all models is shown in table III. As we have learned and explored different techniques of normalization, gradients, optimizers, type of layers and activation function and have applied all models we are confident that we can further take this study to advanced level where we improve these models by using the state of the art optimization techniques. Further we are looking forward to move to semantic segmentation of images and exploring the encoder-decoder architecture for semantic segmenatation of images and implement it in different problem. General architecture of encoder-decoder architecture is shown in Fig. 8.

VII. Conclusions

We have classified multi-class natural scenes into 6 different classes by exploring and implementing the state of the art deep learning models. This study will not only lead us to explore and optimize these models and also implement it in different fields of science. A predictor that can evaluate scenes based on the images, can not only help in the field of urban analysis, object detection, semantic segmentation, agriculture, exploring space and wide range of medical sciences. One of the major applications also includes the self driving cars and safe cities where live image and video feed can be detected by the application in cars, houses and roads for security, safety and automation of traffic system.

Model	Accuracy
Base CNN	54 %
Variant CNN	70 %
VGG16	90 %
DenseNet	84 %
ResNET	91.6 %
InceptionV3	68.9 %

TABLE III: Accuracy Result

References

[1] S. Bergamaschi, F. Guerra, M. Orsini, C. Sartori, and M. Vincini, "A semantic approach to etl technologies," *Data & Knowledge Engineering*, vol. 70, no. 8, pp. 717–731, 2011.

[2] P. Vassiliadis, A. Karagiannis, V. Tziovara, A. Simitsis, and I. Hellas, "Towards a benchmark for etl workflows," 2007.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[5] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," tech. rep., Citeseer, 2009.

[6] A. Fischer and C. Igel, "An introduction to restricted boltzmann machines," in *Iberoamerican congress on pattern recognition*, pp. 14–36, Springer, 2012.

[7] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[8] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Advances in Neural Information Processing Systems*, pp. 901–909, 2016.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[11] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision. arxiv 2015," *arXiv preprint arXiv:1512.00567*, vol. 1512, 2015.

[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Appendix

Code: https://www.kaggle.com/mariamkhalid/image-classification-with-keras
dataset: https://www.kaggle.com/puneet6060/intel-image-classification

Github Link: https://github.com/faisalmaqbool94/Exploratory-Notebooks/blob/master/Coventry.ipynb