Effect of resizing images for Image Classification using Convolutional Neural Networks

Karanraj Dundappa Sankeshwari[1]
B.E., Student,
Department of CSE
SSIT, Tumakuru

Shubham Swaraj[2]
B.E., Student
Department of CSE
SSIT, Tumakuru

Dr.B.H Manjunatha Kumar[3]
Assistant Professor, Department of CSE
Sri Siddhartha Institute of Technology, Tumakuru

Abstract: A lot of progress has been done in the field of image classification with regard to refinements of procedures to train the model as well as with data augmentations and optimizing of methods. In this paper we will observe the effects of scaling images and the accuracy with which the model identifies them. For images we have used the intel image classification database available on Kaggle and pre-processed the images by scaling them down to 30x30,60x60,90x90,120x120, we use a simple CNN to classify the images. The results show the accuracy with which the model identifies these images. We aim to demonstrate that scaling down images have little to moderate effect on the accuracy of detection and that resizing images reduces the time taken for batch processing of images(training and prediction).

I. INTRODUCTION

The convolutional neural networks are deep learning network class which are used to analyse images and are used to as the main backbone in image classification. They are very fast and efficient, vastly used in fields from healthcare to security.

A CNN consists of ReLU layers, Max-pooling layers convolutional layer as well as a full connected layer.

CNNs usually have an input and output layer with hidden layers in between. These hidden layers consist of ReLU, Pooling, Convolutional and pooling layers with some fully connected layers included with them too.

1.Convolutional layer takes input and applies convolutional operations to it , this information is then passed on to the next layer.

2.The output of neurons is then combined together by the Pooling layer into a single neuron present in the next layer..

3. Each neuron in a layer is connected to each neuron in the next layer by fully connected layers.

Input is received to neurons from a small part of the previous layer in the convolutional layer, whereas every neuron receives input from every neuron present in the previous layer in fully connected layer.

The role of the Flatten layer in Keras is super simple: A flatten operation on a tensor reshapes the tensor to have the shape that is equal to the number of elements contained in tensor non including the batch dimension.

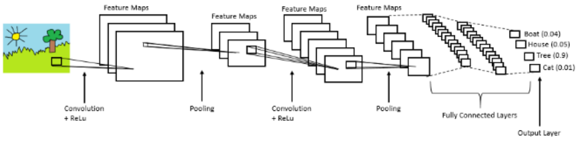


Fig. 1 Basic CNN Network architecture

II. RELATED WORK

Many have presented a working model for the current intel image classification dataset.

We have referred the following implementation provided by Steve Rogers link to the article (https://medium.com/beovolytics/ image-classification-using-keras-intel-scene-classification-challenge-b0b087374654)

The above mentioned article dives deep into how to use pandas ,matplotlib, tensorflow, keras and numpy to classify the images into 6 categories,them being buildings, forest, mountains, glacier, sea, street.the following implementation uses image augmentations provided by keras ImageDataGenrator and a CNN model using ResNet 50 pretrained on ImageNet to achieve an overall accuracy of 91%.

III. DATASET

We use the Dataset provided by Intel.The data here consists of 25,000 images with dimensions 150x150 separated under the following categories.

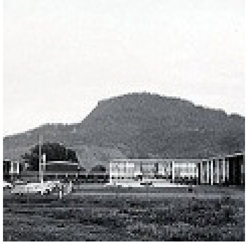
0)Buildings
1)Forest
2)Glacier
3)Mountain
4)sea
5)street

The images are separated into separate zip files consisting of 14000 images in the training folder, 3000 in the testing folder and 7000 in the prediction folder.

The following data was published on https://datahack.analyticsvidhya.com to host a challenge by Intel.

Here in this paper, we preprocess the data by resizing them down to 120x120,90x90,60x60, 30x30.To study the effects on image classifier and the accuracy of the prediction. Resizing any image leads to loss of pixels, our paper tries to analyse whether after loss of certain pixels, the image classifier has issues classifying them into the various classes mentioned above.

Original image



120x120 image

90x90 image

60x60 image

30x30 image

Usually Resizing an image to a bigger size leads to duplication or stretching of certain pixels, and resizing to a smaller size than original leads to loss of data from certain pixels

IV. IMPLEMENTATION

Here we use a simple CNN with

1) 3 Conv2D layers

2) 2 Maxpooling Layers

3) 2 Dense layers one with activation='relu' and another with activation='softmax'

To compile we use the optimizer='adam' and a loss function sparse_categorical_crossentropy the metric used is accuracy

Following are the results for 2 runs each with 12 epochs:

First Run:

Loss=0.7308429479598999

Accuracy=0.7400000095367432

Time=100.50722694

30x30 Image Dataset

Loss=0.5344054698944092

Accuracy=0.8026666641235352

Time=403.54741621

60x60 Image Dataset

Loss=0.5824745297431946

Accuracy=0.7986666560173035

Time=916.378977060

90x90 Image Dataset

Loss=0.667425274848938

Accuracy=0.7903555631637573

Time=1784.9216775894

120x120 Image Dataset

Second Run:

Loss=0.6565883159637451

Accuracy=0.7876666784286499

Time=103.50716641221

30x30 Image Dataset

Loss=0.9343500733375549

Accuracy=0.7976666688919067

Time=407.462364243242

60x60 Image Dataset

Loss=1.181977391242981

Accuracy=0.7913333177566528

Time=914.762672437212

90x90 Image Dataset

Loss=1.4525896310806274

Accuracy=0.7633837461471558

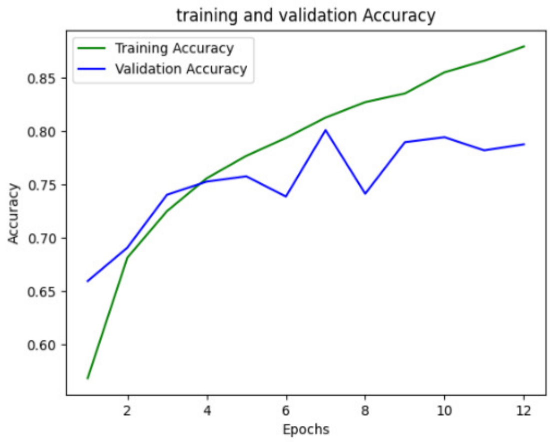
Time=1769.4782347823412

120x120 Image Dataset

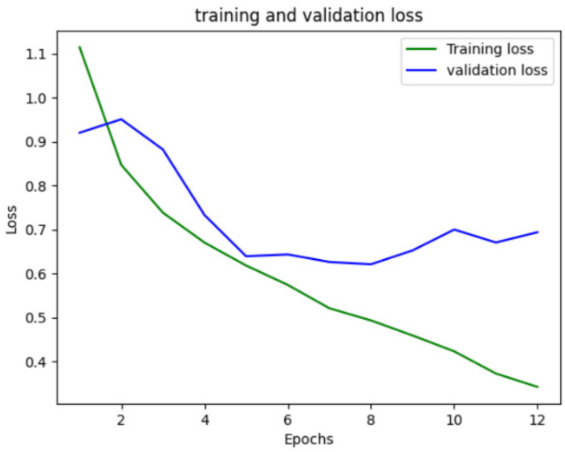
Following are the Training and Validation Loss and Accuracy Graphs

For 30x30 images:

Accuracy:

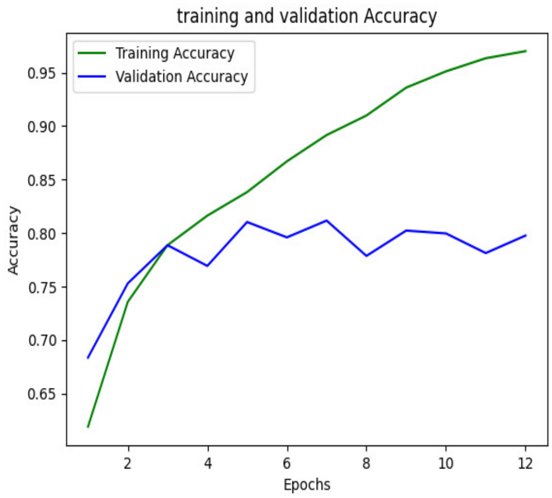


Loss:

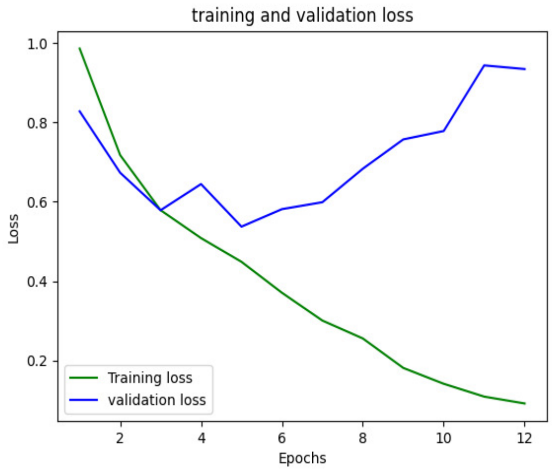


For 60x60 images:

Accuracy:

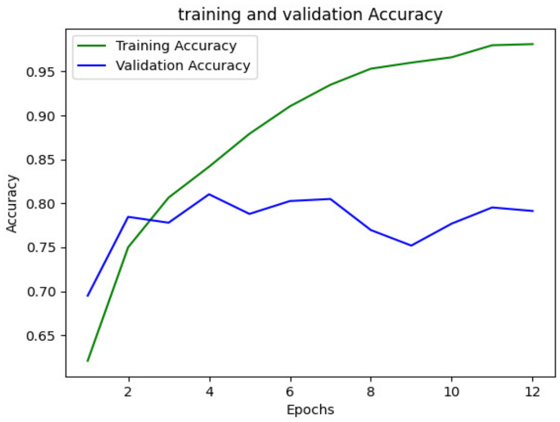


Loss:

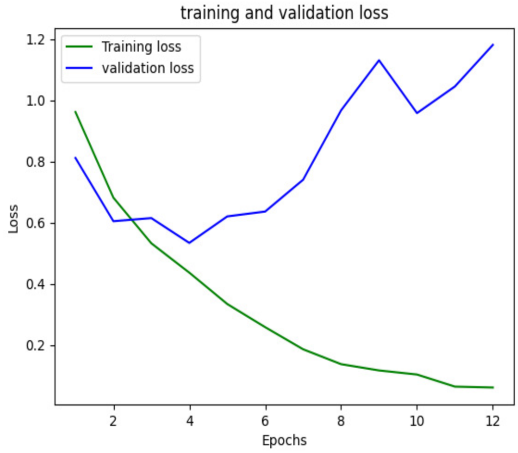


For 90x90 images

Accuracy:

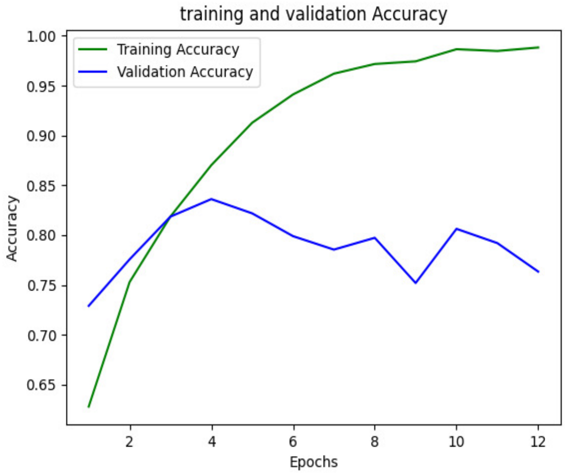


Loss:

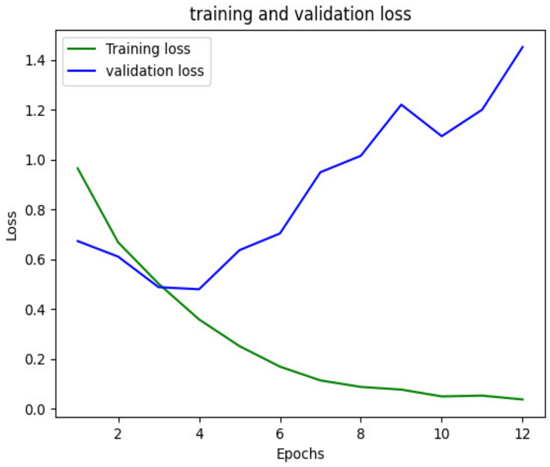


For 120x120 images:

Accuracy:



Loss:



V. CONCLUSION

From the above observed results, there seems to be very small variations in the accuracy with which the images are identified, the model is able to recognise the images accurately on average 75% of the time. This leads us to the conclusion that the data lost during the resizing the images rarely affects the accuracy of prediction.This also has an effect on the training time of the data set, resizing it to a smaller size reduces the time taken by the model to train itself.

REFERENCES

[1] A survey of image classification methods and techniques for improving classification performance-D Lu, Q Weng - International journal of Remote sensing, 2007 - Taylor & Francis.

[2] Bag of tricks for image classification with convolutional neural networks-T He, Z Zhang, H Zhang, Z Zhang…2019 - openaccess.thecvf.com

[3] https://www.youtube.com/watch?v=t0EzVCvQjGE &t=216s-NueraNine