

CS_H10

20191700734	هدى زين العابدين احمد ابراهيم
20191700735	هدى سامي محب عبد الحافظ
20191700733	هبة الله علي سيد محمود
20191700089	أروى علاء الدين عبد الحميد
20191700280	ساره محمود كمال عبدالوهاب

VGG16 Model (pretrained model)

It is consisting of:

- In VGG16 there are thirteen convolutional layers, five Max Pooling layers and three Dense layers which sum up to 21 layers, but it has only sixteen weight layers learnable parameters layer.
- It has convolution layers of 3×3 filter with stride 1 and always used the same padding and maxpool layer of 2×2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture.
- Conv-1 Layer has 64 number of filters, Conv-2 has 128 filters, Conv-3 has 256 filters, Conv 4 and Conv 5 has 512 filters.
- Three Fully-Connected (FC) layers and the final layer is the soft-max layer.
- The input to Conv 1 layer is of fixed size 224×224 RGB image.

Using Transfer Learning:

We are freezing all the convolutional block of VGG16 and last Maxpooling layer it is False which means that during training the parameters of these layers will not be updated and (using pre-trained weights). Whereas the last three layers have trainable parameter sets to true and hence during training the parameter of these layers gets updated.

We use "**categorical_crossentropy**" loss function for multiclass classification and "**ADAM**" optimizer.

Preprocessing:

- Resize images 224×224 .
- Normalization.
- Data augmentation with parameters (featurewise_center=True, featurewise_std_normalization=True, zoom_range=0.15, width_shift_range=0.2, height_shift_range=0.2, shear_range=0.15).
- Changes num of epochs and learning rate.

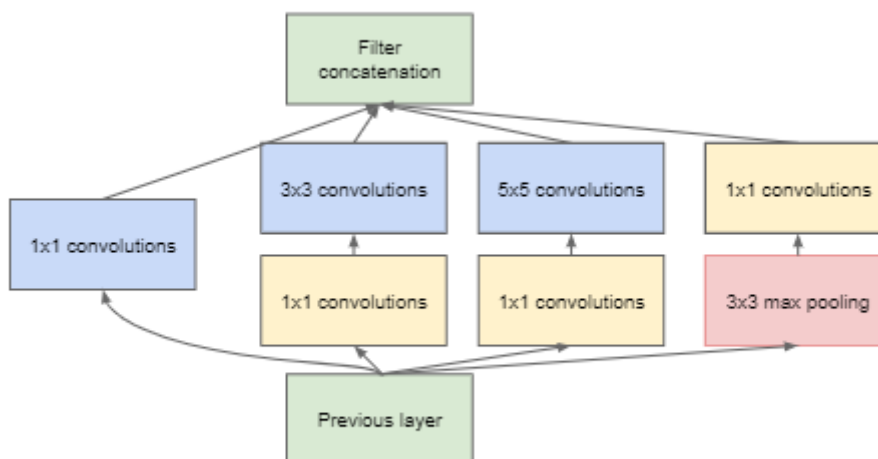
The best training accuracy: 0.9876

The best validation accuracy: 0.9526

GoogleNet Model (NOT pretrained model)

We used different kinds of methods such as 1×1 convolution and global average pooling that enabled it to create deeper architecture. In the architecture, we will discuss some of these methods:

- **1×1 convolution** : The inception architecture uses 1×1 convolution in its architecture. These convolutions used to decrease the number of parameters (weights and biases) of the architecture. By reducing the parameters we also increase the depth of the architecture.
- **Global Average Pooling** : In GoogLeNet architecture, there is a method called global average pooling is used at the end of the network. This layer takes a feature map of 7×7 and averages it to 1×1 . This also decreases the number of trainable parameters to 0 and improves the top-1 accuracy by 0.6%.
- **Inception Module**: In this architecture, there is a fixed convolution size for each layer. In the Inception module 1×1 , 3×3 , 5×5 convolution and 3×3 max pooling performed in a parallel way at the input and the output of these are stacked together to generated final output. The idea behind that convolution filters of different sizes will handle objects at multiple scale better.



(b) Inception module with dimension reductions

- **Auxiliary Classifier for Training**: Inception architecture used some intermediate classifier branches in the middle of the architecture, these branches are used during training only. These branches consist of a 5×5 average pooling layer with a stride of 3, a 1×1 convolutions with 128 filters, two fully connected layers and a softmax

classification layer. These layers help in combating gradient vanishing problem and also provide regularization.

The preprocessing of this model consists of:

- Resize.
- Data Augmentation.
- Normalization.

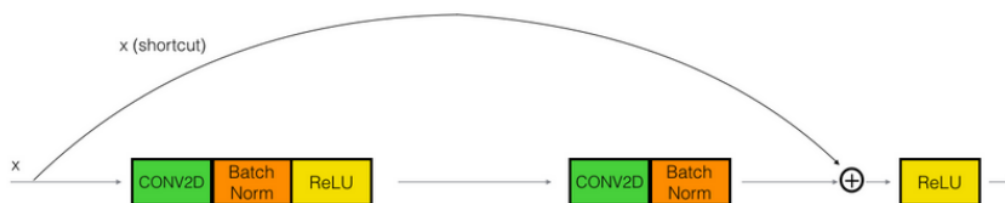
The best training accuracy is 0.95

The best validation accuracy is 0.67

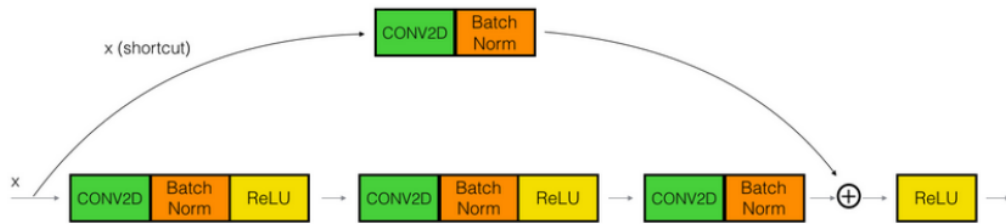
Resnet50 Model (pretrained model)

The ResNet-50 model consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. The ResNet-50 has over 23 million trainable parameters.

- **Identity Block** : The identity block corresponds to the case where the input activation has the same dimension as the output activation.



- **Convolutional Block** : We can use this type of block when the input and output dimensions don't match up. The difference with the identity block is that there is a CONV2D layer in the shortcut path.



The preprocessing of this model consists of 2 types:

- **First preprocessing:**

- ☒ Resize.
- ☒ Data Augmentation.
- ☒ Image sharpening.
- ☒ Normalization.

The training accuracy is 0.97, and the validation accuracy is 0.71.

- **Second preprocessing:**

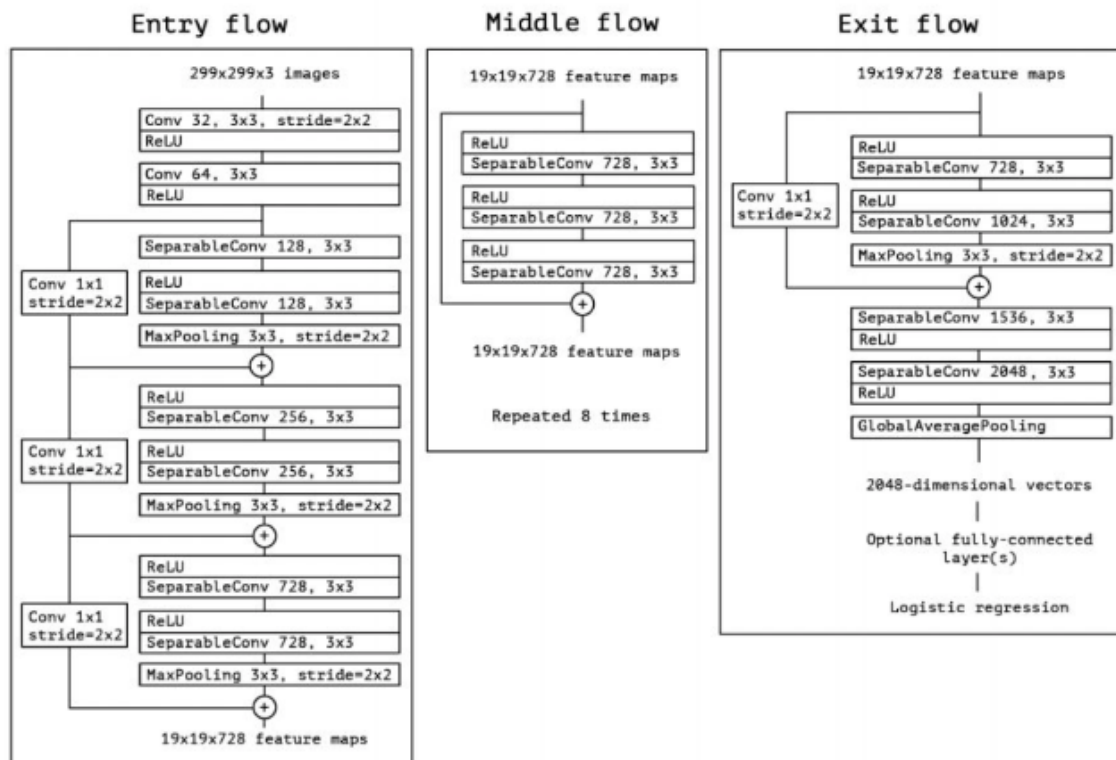
- ☒ Resize.
- ☒ Data Augmentation.

The training accuracy is 0.88, and the validation accuracy is 0.78.

The best training accuracy is **0 . 88** with the second type of preprocessing.

The best validation accuracy is **0 . 78** with the second type of preprocessing.

Xception Model (pretrained model)



Architecture:

Xception is a convolutional neural network. It is the pretrained network. The Xception module has 3 main parts. The Entry flow, the Middle flow (which is repeated 8 times), and the Exit flow.

- The entry flow has two blocks of convolutional layer with 32 and 64 filters respectively. Each followed by a ReLU activation.

There are also various Separable convolutional layers, and Max Pooling layers. we use 'ADD' to merge the two tensors. It begins with an image size of 299x299x3, and after the entry flow, we get an image size of 19x19x728.

- The Middle flow and the Exit flow

the above diagram clearly explains the image size, the various layers, the number of filters, the shape of filters, the type of pooling, the number of repetitions, and the option of adding a fully connected layer in the end.

Also, all Convolutional and Separable Convolutional layers are followed by batch normalization.

- We'll compile our model using categorical crossentropy to reflect the fact that we want to fit into one of many categories, measuring the accuracy of our model, and using Adam as optimizer

Preprocessing:

- Resize image to (229 x 229)
- Normalization

Accuracy:

best accuracy achieved with learning rate=1e-3, and number of epochs =5

the Xception architecture shows much better performance improvement than other models on our dataset with best training accuracy and validation accuracy

training accuracy = 0.9970

validation accuracy = 0.9615