cs577 Assignment 4

Jorge Gonzalez Lopez
A20474413
Department of Computer Science
Illinois Institute of Technology
April 22, 2021

Abstract

Two different classification datasets, one of them binary and the other multi-class have been analysed in order to test different approaches such as visualization of the activations of the layers and of the filters, data augmentation, inception blocks and residual blocks to determine how do they affect the neural network results based on the data selected.

Proposed Solution:

Two different convolutional neural networks have been designed and tuned for the binary and multiclass classification problems, respectively.

For the binary problem the final network has the following design:

Conv(16,3) -> BN -> MaxP(2) -> Conv(32,3) -> BN -> MaxP(2) -> Conv(64,3) -> BN -> MaxP(2) -> Conv(128,3) -> BN -> MaxP(2) -> Conv(256,3) -> BN -> MaxP(2) -> Flatten -> Dense(128) -> DO(0.5) -> Dense(1)

And the multiclass problem as:

Conv(32,3) -> BN -> MaxP(2) -> Conv(64,3) -> BN -> MaxP(2) -> Conv(128,3) -> BN -> Flatten -> Dense(32) -> DO(0.5) -> Dense(10)

Where Conv(n,f) stands for a convolutional layer with n filters with size (f,f), BN for a BatchNormalization layer, MaxP(f) a MaxPooling layer with a kernel size (f,f) and DO(x) a DropOut layer with a x rate.

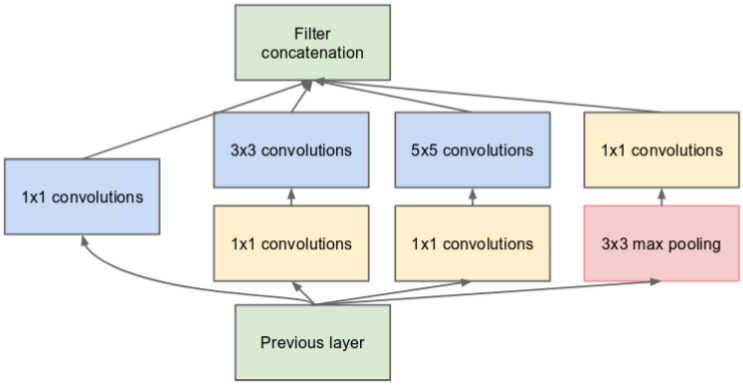
At the same time, the layers of the multiclass problem have been added with a l2 kernel regularization.

Then, once the networks have been defined, the following steps have been done:

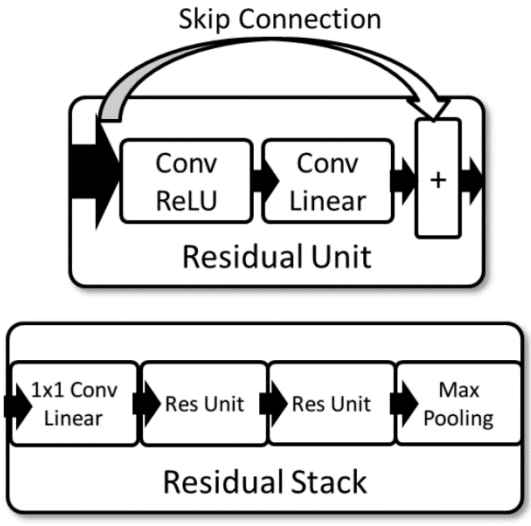
1. Binary classification:
    a. The data selected is just a fraction of the total data available to simulate the condition of limited data.
    b. Visualization of the activations of the layers: One image has been selected and has been passed down to the Model as input. Then, the outputs of the first and third convolutional layers have been plotted.
    c. Visualization of the filters learned: The Keras example of visualizing what convnets learn has been used to find out the inputs that maximize the filters responses. All the filters of the third convolutional layer have been plotted.

d. The VGG16 has been download and used as a pre-trained network: The whole VGG16 model has been used, apart from the last layer, with the weights learned in the imagenet dataset. At the same time, two dense layers have been added at the end of the network. First, the VGG16 has been frozen to train just the added layers, then it has been unfrozen to perform a fine tune and finally, it has been frozen again and retrained with data augmentation.

2. Multi-class classification:
   a. One inception block with dimension reductions has been added to the model with the following structure:



   b. The inception block has been removed and a residual stack (two residual blocks) has been added to the model with the following structure:



Results and Discussion:

1. Binary classification problem:

A subset of 2000 Cat images and 2000 Dog images has been selected to simulate the condition of limited data and those images have been loaded with data generators. The results obtained by training the model are the following:

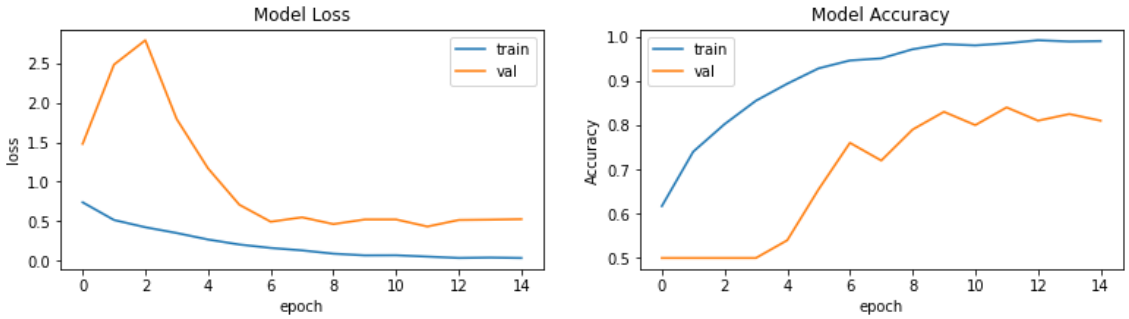


Figure 1: Results of training the model with the limited amount of data.

As it can be seen in the previous graph, as there isn't much data, the model suffers from a little bit of high variance (overfitted) and performs much better on the training dataset.

Then, the activations of the first and third convolutional layers have been displayed:

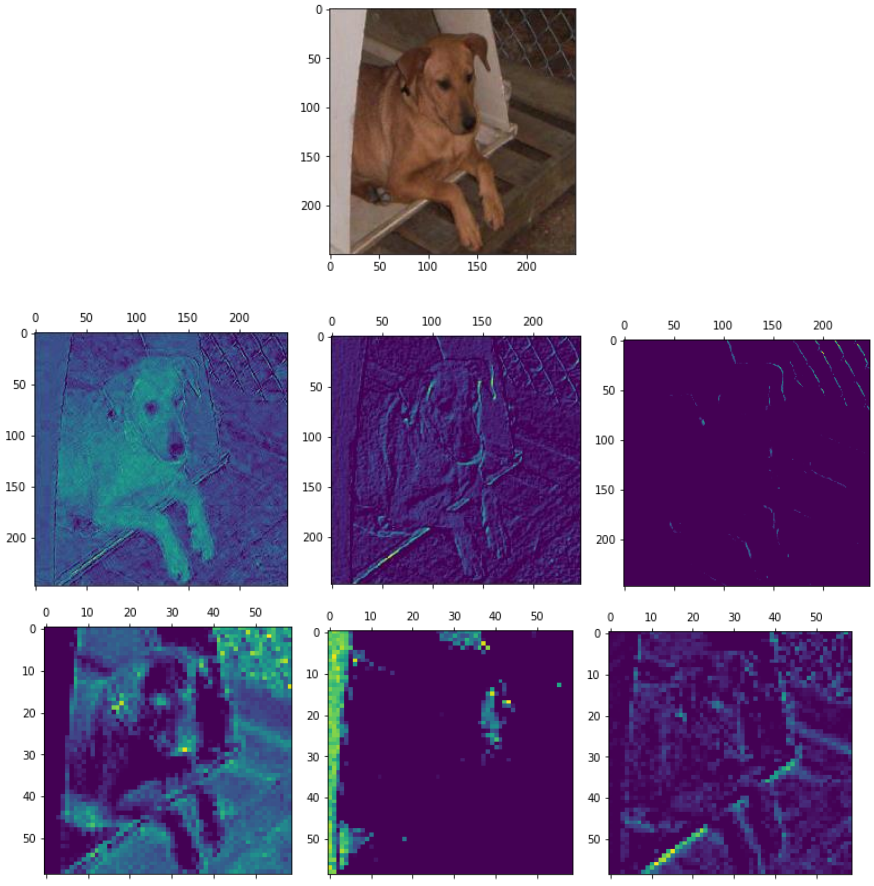


Figure 2: Input image (first row), 3 activations of the first convolutional layer (second row) and 3 activations of the third convolutional layer (third row).

As it can be seen in Figure 2, each filter focuses on different aspects of the images. For example, in Figure 2, some activations show that a filter is detecting edges while others detect color contrast. At the same time, in the third convolutional layer, the filters start focusing on higher level features such as the ear or the mouth of the dog.

Next step, the Keras example of visualizing what convnets learn has been used to find out the inputs that maximize the filters responses. The filters of the third convolutional layer are shown in Figure 3. There, it can be seen that each of the filters represents a particular texture or shape.

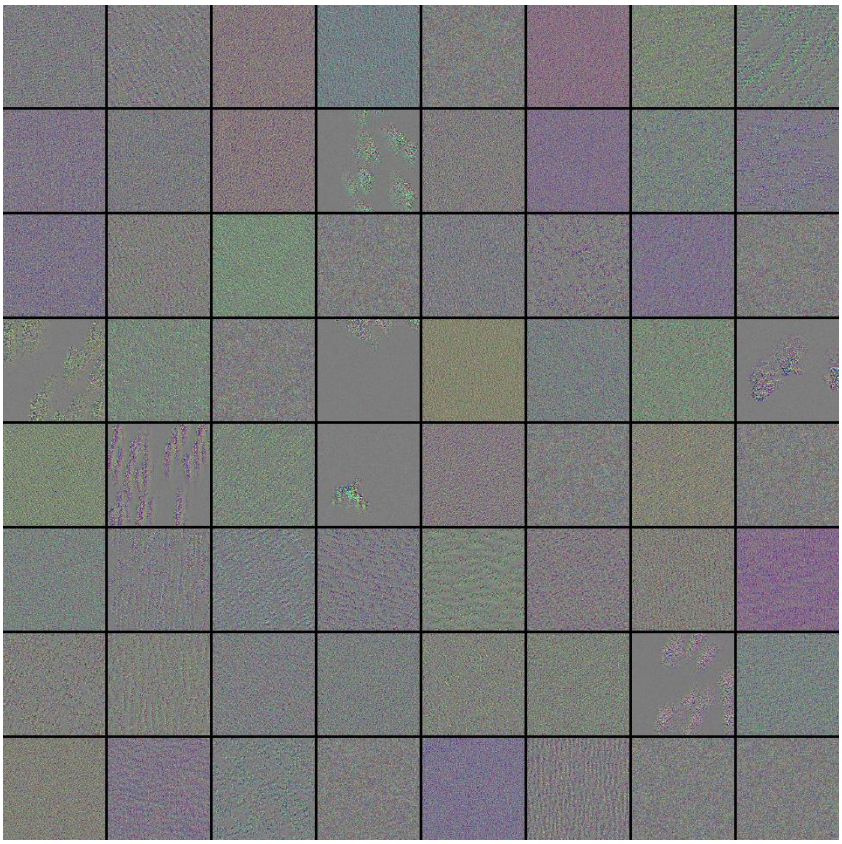


Figure 3: Filters of the third convolutional layer.

Finally, the model has been replaced with the pre-trained convolution base of VGG16, to which a couple of Dense layers have been added to adapt it to the new data. The results obtained when training the model with the convolution base frozen are the following:

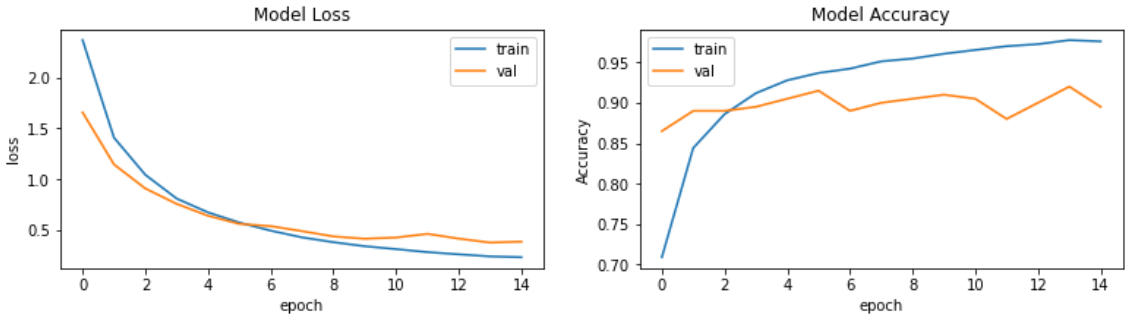


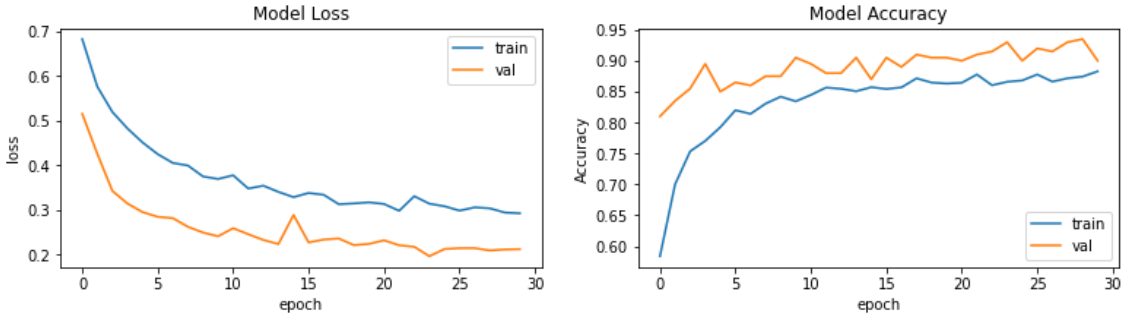
Figure 4: Results after training with the pre-trained model froze.

These results show that as there is not many data to train the model, a pre-traiend one improves the results a lot. Then, the VGG16 convolutional base has been unfrozen and trained again to fine tune the pre-trained parameters. The results obtained can be seen next:

```
Accuracy (VGG16 frozen): 0.9350000023841858
```

```
Accuracy (VGG16 unfrozen): 0.9399999976158142
```

Therefore, fine-tuning the parameters has a little improvement in the total accuracy of the model. Finally, the VGG16 has been frozen again and has been trained with data augmentation with the following results:



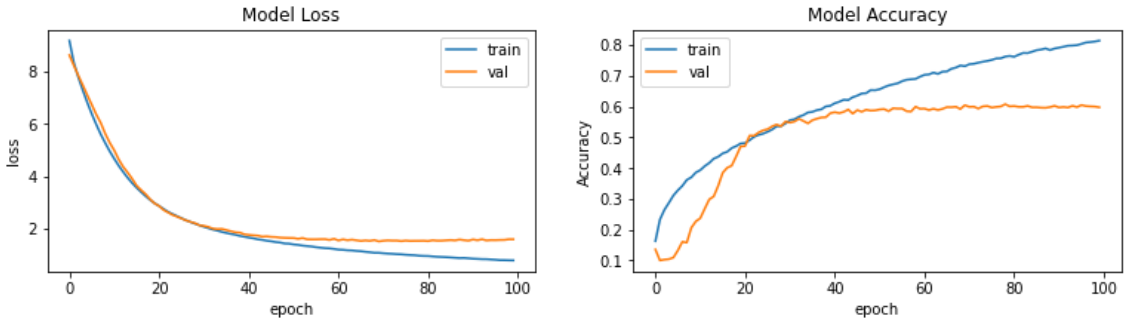

```
Accuracy: 0.949999988079071
```

Figure 5: Results after training with the pre-trained model froze and data augmentation.

These results show that the main problem was the limited data, hence the data augmentation improves the results a lot.

2. Multi-class classification problem:

The images have been loaded with a 32 x 32 x 3 format and have been normalized. Then, the model has been trained with 70% of all the data, adding L2 normalization to the convolutional layers. The results obtained by training the model are the following:

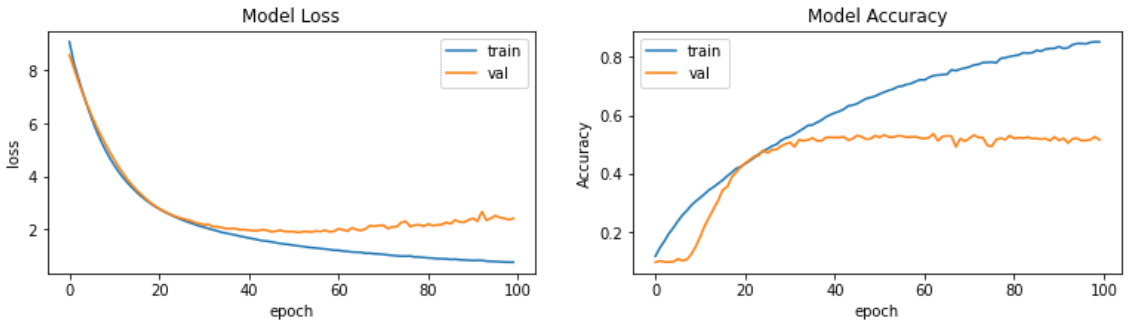



```
loss: 1.6114 - accuracy: 0.5908
```

Figure 6: Results of training the model.

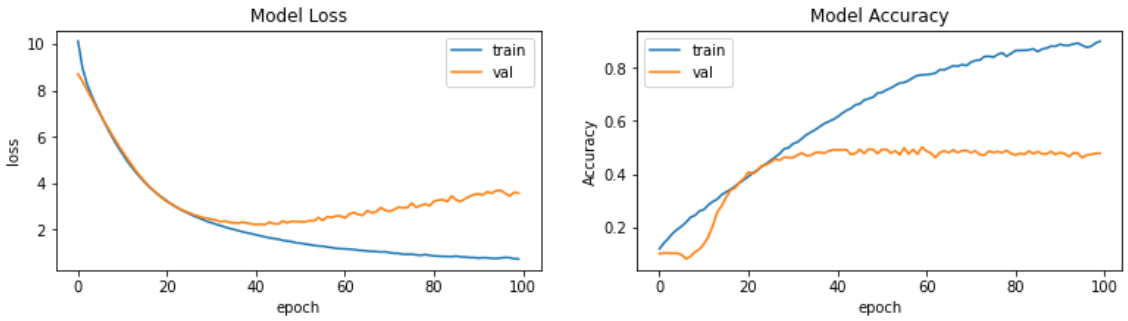
As it can be seen in the previous graph, the model suffers from high variance (overfitted) and performs much better on the training dataset.

Then, one inception block has been added prior to flatten layer as explained in the Section: Proposed Solution. Then, the inception block has been replaced by a residual stack composed of two residual blocks. The results obtained with both of them are shown in Figures 7 and 8, respectively.




```
loss: 2.4385 - accuracy: 0.5089
```

Figure 7: Results of training the model with an inception block.




```
loss: 3.5573 - accuracy: 0.4826
```

Figure 8: Results of training the model with a residual stack.

As it can be seen in the previous graphs, the inception block and the residual stack do not improve the model results. The reason is that the model is already suffering from high variance and therefore, adding more layers to the model (more parameters) will only increase the problem. One possible solution that could improve the results would be replace the first convolutional layers with the inception/residual blocks instead of adding them at the end and increasing the complexity of the network assignment statement was to add them to the previously designed network.