Assignment 2

Convolutional Neural Networks Assignment Report

Cats Vs Dogs:

Aim:

To build a convolutional neural network capable of distinguishing between cat and dog images, and to gain insights into the influence of training dataset size on the model development process.

Building models

I have created 15 models, each with diverse configurations of layers, nodes, optimizers, and various hyperparameter settings.

Classifying the models that were built into two categories: "Scratch Models" and "Pre-Trained Models." The following section presents the hyperparameter tuning details and performance assessments for the models developed from the ground up, along with the subsequent findings derived from these models.

Hyper tuning parameters

Model configuration

Model number	Input Layers	Filters ranges	Filter Size	Optimizer	Training	Validation / Test split	Dropout
Model 1	5	32 to 256	3	Adam	1000	500 & 500	-
Model 2	5	32 to 256	3	Adam	1000	500 & 500	0.5
Model 3	6	32 to 512	3	Adam	1000	500 & 500	0.5
Model 4	5	64 to 1024	3	Adam	1000	500 & 500	0.6
Model 5	5	32 to 256	3	Adam	2000	500 & 500	0.5
Model 6	5	32 to 256	3	Adam	2000	500 & 500	0.5
Model 7	5	32 to 256	3	Adam	3000	500 & 500	0.5
Model 8	5	32 to 256	3	Adam	3000	500 & 500	0.5
Model 9	5	32 to 512	3	Adam	3000	500 & 500	0.5
Model 10	5	32 to 256	3	Adam	5000	500 & 500	0.5

Model performance

Model Number	Max Pooling (Pool size = 2)	Strides (Strides = 2)	Padding	Augmented Images	Test Performance (Loss, Accuracy)
Model 1	Yes	-	-	-	(0.645, 0.614)
Model 2	Yes	-	-	Yes	(0.601, 0.712)
Model 3	Yes	-	-	Yes	(0.609, 0.692)
Model 4	Yes	-	-	Yes	(0.666, 0.652)
Model 5	Yes	-	-	Yes	(0.445, 0.860)
Model 6	-	Yes	-	Yes	(0.608, 0.650)
Model 7	Yes	-	-	Yes	(0.495, 0.818)
Model 8	Yes	Yes	-	Yes	(0.425, 0.856)
Model 9	Yes	Yes	Same	Yes	(0.551, 0.782)
Model 10	Yes	-	-	Yes	(0.182, 0.920)

Results:

- When the training dataset consisted of 1000 samples, the first four models did not achieve notably high accuracy. However, it's important to note that Model 2, which utilized augmented images, exhibited the highest accuracy among these initial models. This observation suggests that data augmentation can be an effective approach to enhance the model's performance.

- In Models 3 and 4, there were changes in the number of layers and the filter configurations. Specifically, Model 3 had 6 layers with filters ranging from 32 to 512, while Model 4 had 5 layers with filters ranging from 64 to 1024. Despite the increase in the number of filters and the addition of layers in these models, it did not lead to an improvement in model performance.
- When the training sample size was increased to 2000 samples for Models 5 and 6, there was a notable improvement in the model's performance, particularly for Model 5. Model 5, which previously achieved 71.2% accuracy with 1000 training samples, exhibited a significant boost in performance when given more training data, achieving an accuracy of 86% and a loss of 44.5%.

- In Model 6, a notable architectural change was made by replacing the pooling layer with a mechanism involving strides to reduce spatial dimensionality. However, this change in the mechanism from pooling to using strides alone did not result in any significant improvement in the

model's performance.

The absence of a pooling layer in Model 6, even with the use of strides, didn't lead to better results. Pooling layers are typically introduced in convolutional neural networks (CNNs) to reduce the spatial dimensions of feature maps and promote translation invariance. They also help in reducing the number of parameters and computation, thus preventing overfitting.

The lack of a pooling layer in Model 6 might have impacted the model's ability to capture and generalize features effectively. In some cases, removing pooling layers without a suitable replacement can lead to issues related to overfitting, as the network may not be able to downsample and extract essential features efficiently.

- Model 7 was again the similar to the previous models 2 and 5 with just the training sample increasing to 3000, it was interesting to observe when the training sample was increased here from 2000 to 3000 the model's accuracy decreased from 86% to 81.8%, here we can again come up with one observation that not just increasing the sample size will improvise the performance, it's key to choose the right sample size so that the model gets to train well and generalize even better on unseen data.
- Model 8 introduced a combination of both Max Pooling and Strides in its architecture. This combination had a positive impact on the model's performance, increasing its accuracy from 81.8% to 85.6%.
- Model 9 was constructed with the use of padding, but this architectural choice did not lead to improved performance. Instead, there was a decline in the model's performance.

When the training sample size was further increased to 5000, the models that had previously exhibited the highest accuracy among different sample sizes, namely Model 2 and Model 5, were employed. This increase in the sample size to 5000 had a significant positive impact on the models' performance.

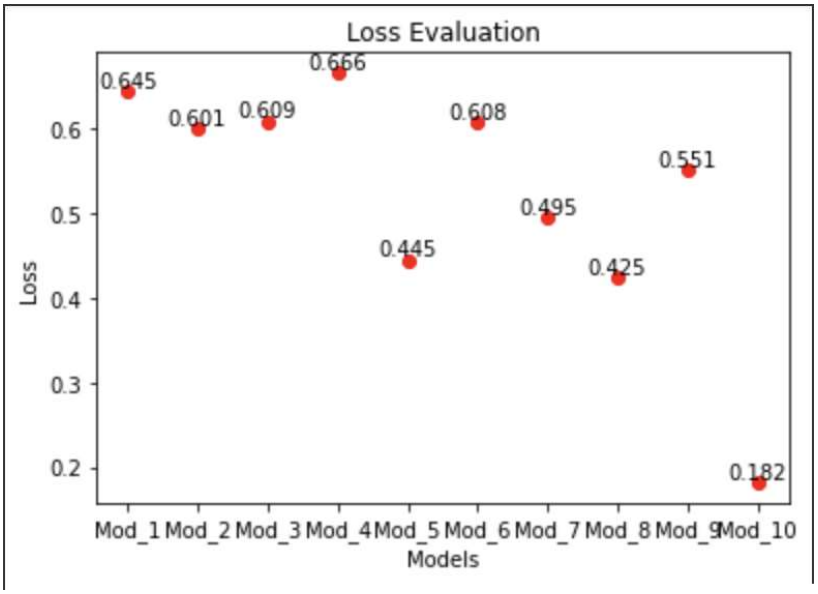
Model 2 and Model 5, which had earlier achieved accuracies of 71.2% and 86%, respectively, with smaller training datasets, experienced a substantial improvement in their accuracy. The accuracy of these models increased to an impressive 92%, with a corresponding loss of 18.2%. This substantial improvement demonstrates the power of having a larger and more diverse training dataset.

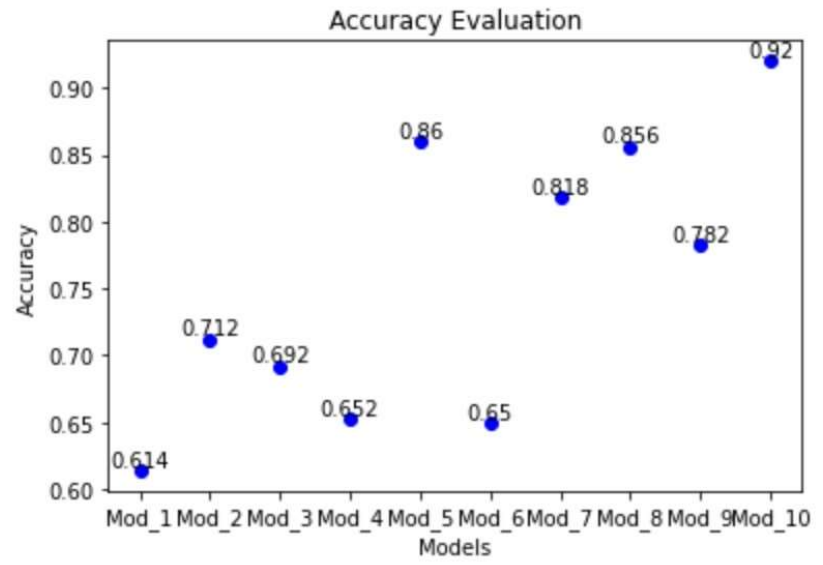
The increase in the training sample size to 5000 allowed these models to generalize better, recognize more patterns, and reduce overfitting, leading to significantly better performance. This underscores the importance of having access to a substantial amount of high-quality training data when training deep learning models, as it can result in substantial gains in accuracy and model effectiveness.

To establish the relationship between the training sample size and the choice of network architecture, it is evident that there is a significant connection between these two factors. The size of the training sample and the design of the network used for training are interrelated and have a substantial impact on the model's performance.

In the analysis, it was observed that the introduction of the Max Pooling operation, along with Dropout and the use of augmented images, was an efficient strategy in most of the models. This strategy contributed to a notable improvement in accuracy and overall model performance. It highlights the importance of selecting the right architectural components and techniques to achieve better results when dealing with different training sample sizes.

The relationship between training sample size and network design underscores the need for careful consideration of both factors when developing deep learning models. The choice of network architecture should align with the available data, and certain architectural elements, such as Max Pooling, Dropout, and data augmentation, can play a significant role in enhancing the model's ability to learn and generalize effectively.




Accuracy Evaluation
Accuracy
Models
0.90
0.85
0.80
0.75
0.70
0.65
0.60
Mod_1
Mod_2
Mod_3
Mod_4
Mod_5
Mod_6
Mod_7
Mod_8
Mod_9
Mod_10
0.614
0.712
0.692
0.652
0.86
0.65
0.818
0.856
0.782
0.92

Pre-trained networks.

- VGG-16, a pre-trained neural network, was used for the image recognition task. VGG-16 is a robust and versatile model that has been trained on an extensive dataset comprising thousands to hundreds of thousands of images across various categories. It is a widely recognized and powerful pre-trained network used extensively around the world for tasks related to image recognition.

5 models were built, out of which 2 of them were the hyper-tuned versions of the previous models.

Hyper tuning pre trained models

Model number	Dense Layer	Training Size	Optimizer	Validation & Test split	Dropout
Model 1	1 (256 Nodes)	1000	rmsprop	500 & 500	0.5
Model 2	1 (256 Nodes)	1000	rmsprop	500 & 500	0.5
Model 3	1 (256 Nodes)	1000	rmsprop	500 & 500	0.5
Model 4	1 (256 Nodes)	5000	Adam	500 & 500	0.5
Model 5	1 (256 Nodes)	5000	Adam (1e-5)	500 & 500	0.5

Model perfomance

Model number	Pre-Trained Weights Update Set to False	Freeze Layers	Augmented Images	Test Performance (Loss, Accuracy)
Model 1	No	False	No	(12.350, 0.956)
Model 2	Yes	False	Yes	(5.111, 0.958)
Model 3	Yes	True	Yes	(9.085, 0.966)
Model 4	Yes	False	Yes	(0.236, 0.986)
Model 5	Yes	True	Yes	(0.083, 0.994)

Results:

Both in the case of pre-trained networks and traditional training approaches, the size of the training sample significantly influenced the model's learning behavior and its subsequent performance on unseen data.

While rmsprop is considered a strong optimizer function for building convolutional neural networks, Adam takes precedence due to its distinct blend of utilizing both Momentum and

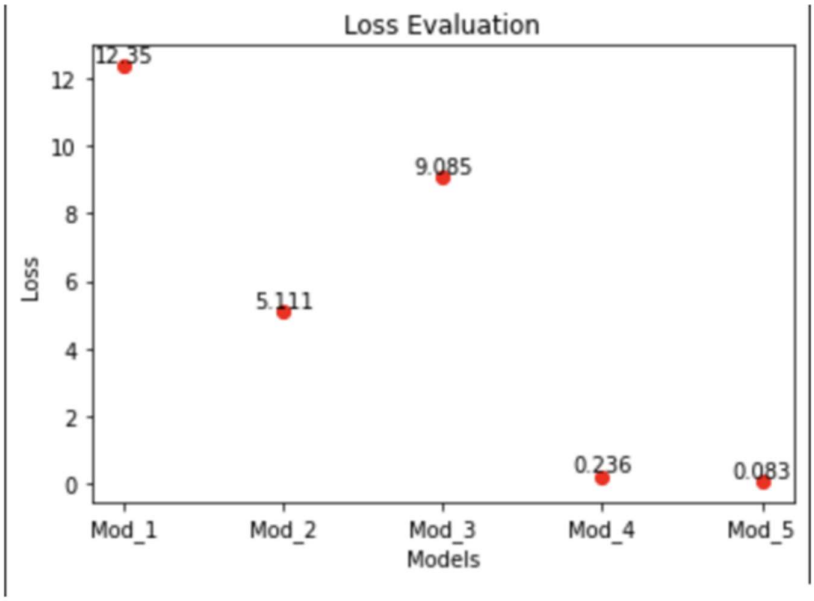
rmsprop techniques in network optimization.

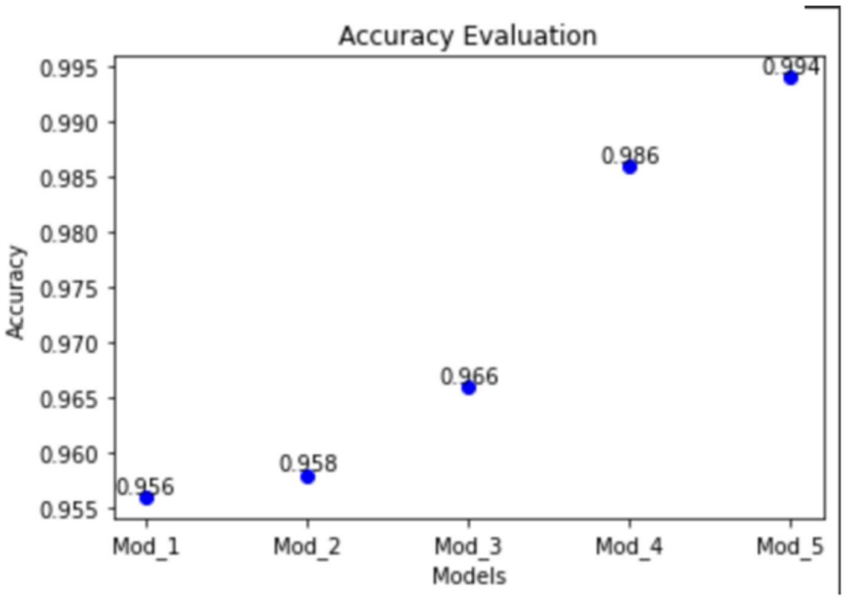
By preventing the pre-trained network from updating its weights, we retain the valuable knowledge embedded in the pre-trained weights. This strategy enables the model to concentrate on training the densely connected classifier layer at the end, leading to notable improvements in the performance of Models 2 and 3. Moreover, it serves as an effective technique to mitigate the risk of overfitting.

pre-trained networks are originally trained on a wide range of image categories, the act of freezing the initial layers essentially locks the general categorization knowledge. This compels the model to concentrate on the finer nuances of image recognition specific to the task at hand. This approach has yielded favorable results, as evidenced by the high accuracy achieved in fine-tuned models, such as Model 3 and Model 5, which are the top performers within their respective sample size categories.

Even in the context of pre-trained networks, the sample size exhibits a significant relationship. When transitioning from a sample size of 1000 to 5000, a noticeable improvement in the model's performance became evident. This augmentation of the sample size resulted in reduced loss values, indicating more precise predictions.

Providing the model with a larger dataset and incorporating augmented versions of the images proved to be an efficient strategy to enhance the model's accuracy. Additionally, freezing the initial layers of the pre-trained network and preventing it from updating its weights during training emerged as effective methods to prevent overfitting and promote robust generalization to unseen data.


Loss Evaluation
12.35
9.085
5.111
0.236
0.083
Loss
0
2
4
6
8
10
12
Mod_1
Mod_2
Mod_3
Mod_4
Mod_5
Models


Accuracy Evaluation
0.994
0.986
0.966
0.958
0.956
Accuracy
0.955
0.960
0.965
0.970
0.975
0.980
0.985
0.990
0.995
Mod_1
Mod_2
Mod_3
Mod_4
Mod_5
Models