Employing Resnet to Categorize Animal Images

Mohammad Mohammad Beigi[a]

[a] Student, Department, Sharif University of Technology

This manuscript was compiled on February 7, 2024

Feature extraction in deep learning involves using pre-trained convolutional neural networks (CNNs) like VGG, ResNet, or MobileNet. These networks learn image representations from large datasets. To extract features, one can remove the fully connected layers from the CNN, using only the convolutional layers. The output of these layers can be utilized for different tasks like classification or object detection, known as transfer learning. Alternatively, autoencoders can also be used to learn efficient data representations. In this homework, a ResNet model is employed to extract features from a dataset of dog and cat images for image categorization.

Keywords: Hoda Dataset | sumfilter | maxfilter | SVM | d'

Feature extraction in the realm of deep learning is typically done through the use of pre-trained convolutional neural networks (CNNs), such as VGG, ResNet, or MobileNet. These networks are trained on large datasets to learn hierarchical representations of images.

To perform feature extraction using a pre-trained CNN, one can remove the fully connected layers at the top of the network, leaving only the convolutional layers. The output of these convolutional layers can then be used as features for a different task, such as classification or object detection. This process is often referred to as transfer learning, where the knowledge gained from one task is transferred to a different, but related, task.

Another approach to feature extraction in deep learning involves using autoencoders, which are neural network models designed to learn efficient representations of the input data. Once trained, the hidden layers of the autoencoder can be used as features for downstream tasks.

In this report we use a Resnet and a VGG model to extract features of a dataset containing dog and cat images. The aim is to categorize the images.

Materials and Methods

Dataset. Our Dataset includes 2000 train images and 1000 test images. Half of the images are dogs and half of them are cats.

Fig. 1. Dataset Images

Resnet. ResNet101 is a convolutional neural network architecture that is 101 layers deep. It is known for its use of residual connections, which allow for the training of very deep networks effectively without suffering from the vanishing gradient problem.

Residual connections alleviate the degradation problem by allowing the network to learn the difference between the actual output and the output that would be expected if the layers were ideal, making it easier to optimize.

ResNet101 has been shown to achieve state-of-the-art performance in various computer vision tasks, including image classification, object detection, and image segmentation.

It is known for its exceptional ability to capture high-level features from images, making it popular in the field of image understanding and analysis.

Its design makes it easier to train deep neural networks, and it has become a popular choice in deep learning research and applications.

VGG. VGG16, or the Visual Geometry Group 16-layer model, is a deep convolutional neural network architecture designed for image classification tasks. Developed by the Visual Geometry Group at the University of Oxford, it gained prominence for its simplicity and effectiveness. The network consists of 16 layers, including 13 convolutional layers and three fully connected layers. The convolutional layers use small 3x3 filters with a stride of 1, allowing the model to learn intricate patterns and features in the input images.

One notable characteristic of VGG16 is its uniform architecture, where the convolutional layers are stacked consecutively, maintaining a consistent filter size and stride throughout. This simplicity contributes to its ease of understanding and implementation. The network architecture allows for a deeper understanding of hierarchical features in images, enabling better generalization to various visual recognition tasks. Despite its success, VGG16 has been surpassed by more recent architectures like ResNet and EfficientNet, which achieve comparable or better performance with significantly fewer parameters, addressing some of the computational challenges associated with deep networks.

Statistics.

F-score. It is a measure of a test's accuracy. It considers both the precision and recall of the test to compute the score. The F1 score reaches its best value at 1 and worst at 0.

Accuracy. It is the measure of correct predictions made by the model out of all the predictions made. It's calculated by dividing the number of correct predictions by the total number of predictions.

TPR (True Positive Rate). Also known as sensitivity or recall, it measures the proportion of actual positive cases that were correctly identified by the model.

FPR (False Positive Rate). It measures the proportion of actual negative cases that were incorrectly classified as positive by the model.

AUC (Area Under the Curve). It is a measure used to evaluate the performance of a binary classification model. The AUC represents the degree or measure of separability, i.e., how well the model is capable of distinguishing between classes. An AUC value closer to 1 indicates a better model performance.

Classifiers.

MLP (Multi-Layer Perceptron). The MLP is a type of feedforward artificial neural network. It consists of multiple layers of nodes, each connected to the nodes in the adjacent layers. The MLP is trained using supervised learning and is capable of learning non-linear relationships in the data. It can be used for both classification and regression tasks.

SVM (Support Vector Machine). SVM is a supervised learning algorithm used for classification and regression analysis. In the context of classification, SVM finds the optimal hyperplane that best separates the data into different classes. It works by mapping input data to a high-dimensional feature space and finding the hyperplane that maximizes the margin between classes. SVM can also handle non-linear relationships using techniques such as the kernel trick.

Random Forest. Random Forest is an ensemble learning method that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes in the individual trees. Each tree in the random forest operates independently, and the final classification is determined by combining the results from all the trees.

Gaussian Noise. To add Gaussian noise to an image, each pixel in the image is modified by adding a random value drawn from a Gaussian distribution. The process of adding Gaussian noise can be described as follows:

1. Generate random numbers: For each pixel in the image, generate a random number from a Gaussian distribution with a specified mean and standard deviation. The mean controls the average value of the noise, while the standard deviation controls the spread or intensity of the noise.

2. Add noise to the pixel values: Add the generated random numbers to the corresponding pixel values in the image. This effectively perturbs the pixel values with random noise.

Mathematically, if I(x, y) represents the intensity of the pixel at position (x, y) in the original image, and N(x, y) represents the Gaussian noise value at the same position, the pixel value in the noisy image, I'(x, y), can be computed as:

$$I'(x, y) = I(x, y) + N(x, y)$$

Adding Gaussian noise to images can be useful for various purposes, such as simulating the effect of noise in real-world imaging conditions, testing the robustness of image processing algorithms, or augmenting datasets for training machine learning models.

Fig. 2

Salt and Pepper Noise. To add salt and pepper noise to the Persian digital dataset, we can apply the following formula to each pixel in the image:

$$f(x, y) = \begin{cases} 0 & \text{with probability } P \\ 1 & \text{with probability } P \\ I(x, y) & \text{otherwise} \end{cases}$$

Where $I(x, y)$ represents the original intensity of the pixel, and P is the probability of the pixel being corrupted. By applying this formula, we can simulate the effect of salt and pepper noise on the Persian digital dataset.

Results

The results of Resnet101 per different models are shown in Table 1. As you see SVM performance is better than other models. Also you can observe confusion matrices of each classification task.

Model	Accuracy	F Score	AUC
SVM	0.9920	0.9920	0.9998
MLP	0.9890	0.9890	0.9997
Random Forest	0.9840	0.9841	0.9996

Table 1. Performance of Resnet for Different Models

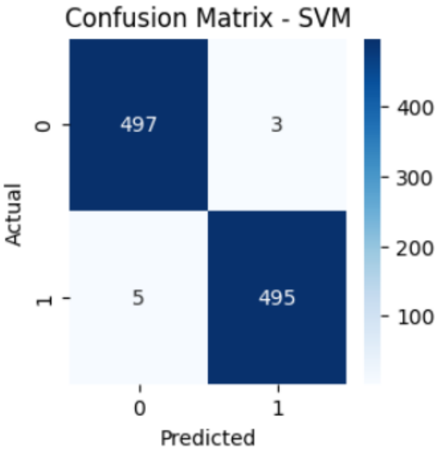

Fig. 3. SVM Confusion Matrix(0 : cat , 1 : dog) - Resnet101

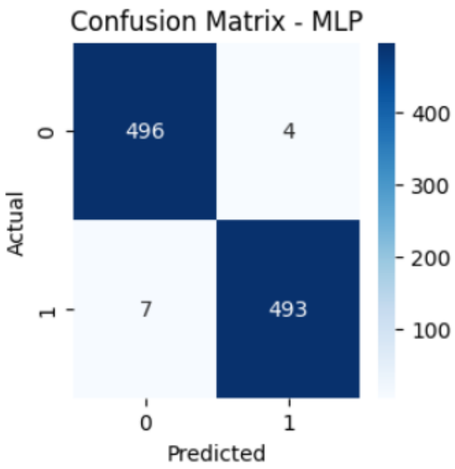

Fig. 4. MLP Confusion Matrix - Resnet101

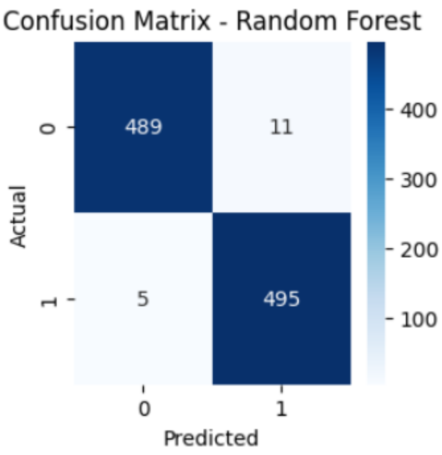

Fig. 5. Random Forest Confusion Matrix - Resnet101

In the SVM confusion matrix, there were 497 true predictions for "cat", 495 true predictions for "dog", 5 false predictions for "cat", and 3 false predictions for "dog".

In the MLP confusion matrix, there were 496 true predictions for "cat", 493 true predictions for "dog", 7 false predictions for "cat", and 4 false predictions for "dog".

In the Random Forest confusion matrix, there were 489 true predictions for "cat", 495 true predictions for "dog", 5 false predictions for "cat", and 11 false predictions for "dog".

In another scenario we have added gaussian noise with std = 40 to the data.

Model	Accuracy	F Score	AUC
SVM	0.8590	0.8753	0.9939
MLP	0.9230	0.9277	0.9949
Random Forest	0.8770	0.8895	0.9952

Table 2. Performance of Different Models on Noisy Test Data

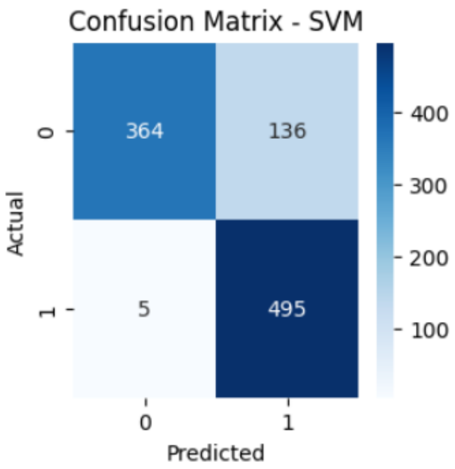

Fig. 6. SVM Confusion Matrix - Resnet101

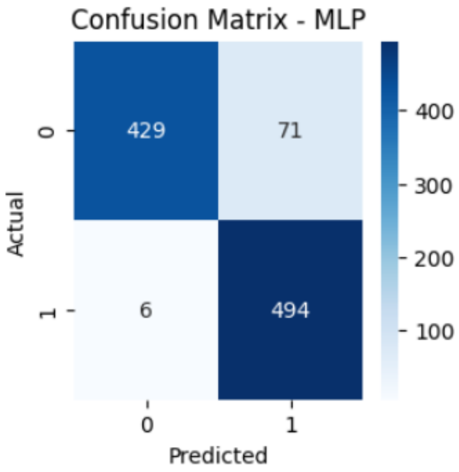

Fig. 7. MLP Confusion Matrix - Resnet101

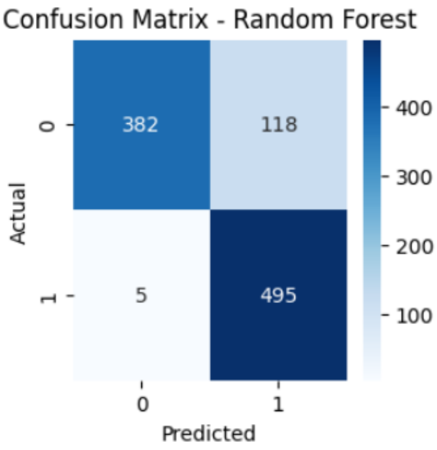

Fig. 8. Random Forest Confusion Matrix - Resnet101

Based on these results, it looks like the MLP algorithm performed the best in terms of accuracy, F score, and AUC. It achieved the highest accuracy of 0.9230, followed by Random

Forest with an accuracy of 0.8770, and SVM with an accuracy of 0.8590. The AUC is also highest for the MLP model, indicating better overall performance in terms of distinguishing between positive and negative cases. The confusion matrices provide a detailed breakdown of the model's performance on individual classes.

Overall, in noisy test data it seems like the MLP model outperformed the other two models based on the given metrics.

The results of VGG16 per different models are shown in Table 2. As you see SVM performance is better than other models. Also you can observe confusion matrices of each classification task.

Model	Accuracy	F Score	AUC
SVM	0.9910	0.9910	0.9996
MLP	0.9800	0.9800	0.9987
Random Forest	0.9800	0.9801	0.9988

Table 3. Performance of VGG for Different Models

The analysis is same as Resnet101.

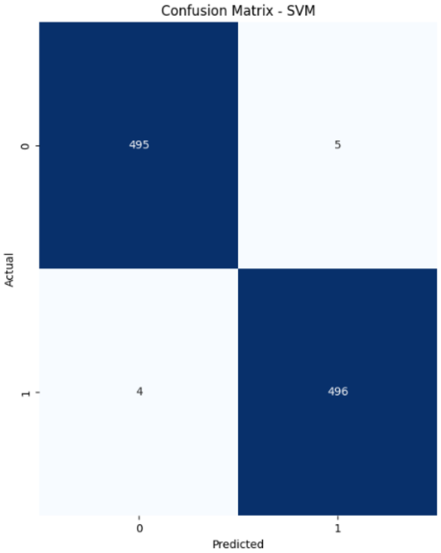

Fig. 9. SVM Confusion Matrix - VGG16

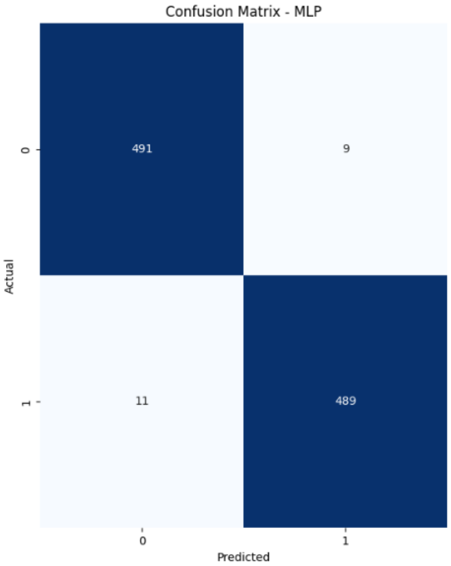

Fig. 10. MLP Confusion Matrix - VGG16

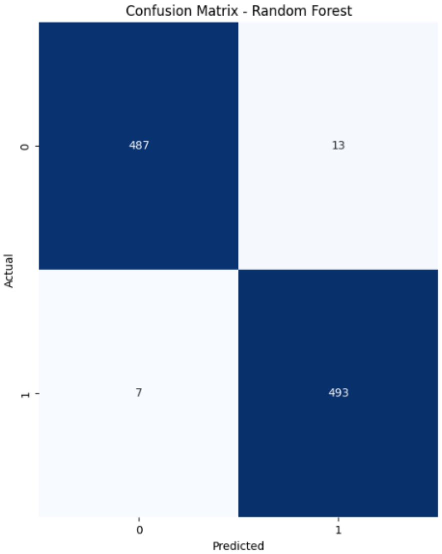

Fig. 11. Random Forest Confusion Matrix - VGG16

Conclusion

Based on the experiment, it is observed that the model, which comprises a ResNet101 or VGG16 feature extractor and different classifiers (SVM, MLP, random forest), demonstrates close accuracies when evaluated with original test images. However, the SVM classifier outperforms the other classifiers in terms of accuracy.

Moreover, when noisy test data is used instead of the original test data, the overall accuracy decreases. Interestingly, in this scenario, the MLP classifier performs significantly better than the others. This suggests that the MLP classifier may be more robust or better able to handle noisy data compared to the other classifiers.

In conclusion, while the SVM classifier yields the best overall accuracy with original test images, the MLP classifier shows promise for handling noisy data more effectively. Further investigation into the performance of these classifiers with noisy data could provide valuable insights for future applications.

References

1. Deep Residual Learning for Image Recognition; Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun