ASSIGNMENT -03
CONVOLUTION

Summary:

Convolutional neural networks, often known as convnets, are a well-liked class of deep learning models that have excelled in computer vision applications. Convnets' capacity to recognize and learn from spatial patterns in pictures is one of its main advantages. They are therefore ideally suited for jobs like segmentation, object identification, and picture recognition.

We think that despite the sparse data, our convnet model can still generate accurate findings. This is a result of convnets' capacity to learn and generalize from tiny datasets by extracting and detecting pertinent characteristics from pictures. We want to train our model on the little dataset, then refine it utilizing transfer learning strategies and assess its performance using suitable evaluation criteria. Our overall objective is to create an accurate and effective convolutional neural network that can successfully categorize photos from the "Dog-vs-Cats" dataset with a small quantity of input.

Purpose:

Create a convolution network using a dataset of images of cats and dogs so that the model can learn the parameters of these images and correctly identify unknown images (test data) as cats and dogs.

Dataset:

The "Dog-vs-Cats" dataset that is published on Kaggle is the source of the dataset we are using, The Cats-vs-Dogs dataset is 543MB in size and comprises 25,000 images of dogs and cats, 12,500 of each type (compressed), We need to expand the capacity of our neural network because the topic we are working on is more difficult and has larger images. To do this, we will extend the Conv2D + MaxPooling2D design by one more level. By lowering the size of the feature maps, this will increase network capacity while also guaranteeing that they are manageable when we get to the Flatten layer. The size of our input photos is originally 150x150, and as we go through the network layers, the feature maps steadily get smaller until they are 7x7 just before the Flatten layer. Although rather random, this choice of input size is suitable for the case.

Data Augmentation:

We want to apply data augmentation approaches to improve the precision of our model. By producing new data from the current training samples through random changes, data augmentation helps us to get excellent results even with small datasets. As a consequence, the model will never see the same image again during training, which enhances its generalization capacity.

We want to randomly resize, flip, rotate, and zoom the photos in the training data for our particular assignment. By doing this, we may alter the current photos, broadening the dataset's variety and enhancing the model's resilience.

Pre trained network:

We may benefit from the pre-trained model's feature extraction skills while employing a pre-trained network because it has previously mastered recognizing a variety of widespread visual properties,The weights of the entire network or just a selection of layers may then be adjusted once we put our custom classifier on top of the previously trained model.

We use pre-trained princes with data augmentation, then optimize the model for best results. With the present dataset, which consists of 1.4 million annotated pictures and 1000 distinct classes, we employ the ImageNet dataset trained on the VGG16 architecture.

Feature Extraction: To extract features from the photos, I'll utilize the VGG16 network that has already been trained. The final layer of the network that is entirely connected will be removed, and a new layer that is fully connected will take its place. Then, I will train just this layer on the Cats & Dogs dataset while freezing the weights of the other levels in the VGG16 network aside from the last fully connected layer.

Fine tuning: I will adjust the pre-trained VGG16 network by unfreezing some of the final convolutional layers and the fully connected layer created in Step 2. Following that, the network will be trained using the Cats & Dogs dataset.

Result:

TABLE FOR MODEL:

Train Size	**Test Size**	**Validation Size**	**Data Augmentation**	**Train Accuracy**	**Validation Accuracy**
1000	500	500	NO	77.4%	71.2%
1000	500	500	YES	73.2%	73%
1500	500	500	NO	84.5%	73.6%
1500	500	500	YES	73.1%	70.9%
1500	1000	500	YES	85.6%	73.7%
1500	1000	500	NO	53.7%	54.5%

TABLE FOR PRE TRAINED MODEL:

Data Augmentation	**Train Accuracy**	**Validation Accuracy**
NO	99.8%	97.3%
YES	96.9%	97.4%

Conclusion:

we provide outcomes for models trained with increased train size or with various train and validation sizes, as well as outcomes with and without data augmentation. We compare the accuracy, validation accuracy, and data augmentation for the pre-trained model.

The results show that the models trained with data augmentation consistently did not outperform those trained without it. Also, modifying the size of the validation set or expanding the training set increases the model's accuracy. When we compare the pre-trained model with and without data augmentation, we can observe that data augmentation did not result in an increase in the accuracy or validation accuracy of the model, Pre-trained models often perform better overall than models that are created from scratch, especially when coping with a dearth of training data.