# Assignment - 2
# COVOLUTION
**Meghana Udiga**

**Executive Summary:**

The aim of our research is to build a new convolutional neural network from scratch, specifically designed for computer vision applications. The dataset that we are utilizing comes from the "Dog-vs-Cats" dataset that is released on Kaggle. The limited data we have makes it challenging to build an effective model.

Convolutional neural networks, sometimes referred to as convnets, are a popular type of deep learning models that have demonstrated superior performance in computer vision applications. One of Convnets' key benefits is its ability to spot and comprehend spatial patterns in images. As a result, they are perfect for tasks like segmentation, object recognition, and image recognition.

We believe that despite the limited data, our convnet model can still produce reliable results. This is because convnets can learn and generalize from small datasets by extracting and recognizing relevant properties from images. The tiny dataset will be used to train our model, which will then be improved utilizing transfer learning techniques. Finally, the performance of our model will be confirmed using appropriate performance metrics. Our main goal is to build a convolutional neural network that can efficiently and effectively classify images from the "Dog-vs-Cats" dataset using a little amount of training data.

**Problem:**

The goal of the Cats-vs-Dogs dataset, a binary classification problem, is to predict whether a photograph belongs to the dog class or the cat class.

**Techniques:**

**Dataset:**

The Cats-vs-Dogs dataset has 25,000 pictures of dogs and cats, 12,500 of each kind, and is 543 MB in size (compressed). We will split it into three subsets when it has been downloaded and uncompressed: a training set with 1000 samples of each class, a validation set with 500 samples of each class, and finally a test set with 500 samples of each class. Due to the larger picture size and more difficult nature of the issue we are working on, we need to increase the neural network's capacity.

To do this, we will add one stage to the present Conv2D + MaxPooling2D architecture. This will reduce the size of the feature maps and increase network bandwidth, resulting in smaller feature maps when we reach the Flatten layer. The feature maps gradually grow smaller as we move through the network layers until they are 7x7 immediately before the Flatten layer. Our input images are 150x150 pixels in size. Although the choice of input size is rather arbitrary, it is appropriate in this specific instance.

**Preprocessing:**

- Read the picture files.
- Decode the JPEG content to RBG grids of pixels.
- Convert these into floating point tensors.
- Rescale the pixel values (between 0 and 255) to the [0, 1] interval (as you know, neural networks prefer to deal with small input values).

**Data Augmentation:**

To increase the model's precision, we want to use data augmentation techniques. By generating more data from the present training samples through random adjustments, data augmentation enables us to get outstanding outcomes even with small datasets. As a result, the model will never view the same image twice during training, increasing its potential for generalization.

In order to perform our specific work, we want to erroneously flip, rotate, and zoom the images from the training set. This will increase the diversity of the dataset and strengthen the robustness of the model by changing the already-existing photographs.

**Pre-trained model:**

Assuming the initial dataset was substantial and varied, a pretrained network may be utilized as a generic model and its properties may be applied to a number of different computer vision applications. Deep learning's ability to transfer learned traits across different tasks is one of its key benefits over other machine learning techniques.

Let's look at a large convolutional neural network as an example, trained on the ImageNet dataset, which consists of 1.4 million annotated images and 1,000 different classifications. Many animal species, including several cat and dog breeds, are featured in this collection. The architecture of this network is a basic and well-liked convnet architecture for ImageNet, known as VGG16.

There are two methods to use a pretrained network: feature extraction and fine-tuning. We will first do feature extraction without data augmentation and then feature extraction with data augmentation in order to achieve even better results.

**Results:** The table below shows each strategy's accuracy and validation loss.

**Table For Model from Scratch:**

| **Train Size** | **Test Size** | Validation Size | **Data Augmentation** | **Train Accuracy(%)** | **Validation Accuracy(%)** |
|---|---|---|---|---|---|
| 1000 | 500 | 500 | NO | 79.05 | 71.20 |
| 1000 | 500 | 500 | YES | 69.05 | 68.80 |
| 1500 | 500 | 500 | NO | 87.17 | 72.20 |
| 1500 | 500 | 500 | YES | 71.97 | 67.20 |
| 1500 | 1000 | 500 | YES | 84.70 | 68.80 |
| 1500 | 1000 | 500 | NO | 69.07 | 71.65 |

**Table For Pre-Trained Model:**

| **Data Augmentation** | **Train Accuracy(%)** | **Validation Accuracy(%)** |
|---|---|---|
| **NO** | 99.87 | 97.40 |
| **YES** | 96.70 | 97.30 |

In the tables above, along with the model settings, are the sample sizes for the train, test, and validation sets. The model built totally includes results with and without data augmentation, for models trained with an increase in train size, or for models trained with varied train and validation sizes. We compare the pre-trained model's accuracy, validation accuracy, and accuracy with and without data augmentation.

The findings demonstrate that models that often underwent data augmentation during training were unable to outperform those that did not. By altering the size of the validation set or the training set, the model's accuracy is also improved. We can observe that the pre-trained model's accuracy and validation accuracy did not improve as a consequence of the data augmentation by comparing it to the pre-trained model without it. Pre-trained models frequently outperform models that are built from scratch overall, particularly when dealing with a lack of training data.