

**Department of Electrical and Computer Engineering**  
**North South University**

---



**Nearest Neighbour of Different Image  
Representation**

**Final report**

**Submitted by:**

**Ankur Chowdhury ID:1911844042**

**Sajid Wasif ID: 1912313642**

**Faculty Advisor:**

**Dr. Mohammad Ashrafuzzaman Khan**

**Assistant Professor**

**Fall 2022**

# Table of Contents

<b>Chapter 1: Introduction</b>	<b>3</b>
<b>1.1 What is Image to Vector?</b>	<b>3</b>
<b>1.2 What is Nearest neighbor of an image?</b>	<b>5</b>
<b>1.3 What is Euclidean distance?</b>	<b>6</b>
<b>1.4 Cosine similarity</b>	<b>7</b>
<b>1.5 Motivations</b>	<b>7</b>
<b>1.6 Aims and Objectives</b>	<b>7</b>
<b>Chapter 2: Literature Review</b>	<b>8</b>
<b>2.1 Related Paper-1</b>	<b>8</b>
<b>2.2 Related Paper-2</b>	<b>8</b>
<b>2.3 Related Paper-3</b>	<b>9</b>
<b>Chapter 3: Methodology</b>	<b>10</b>
<b>3.1 Workflow</b>	<b>10</b>
<b>3.2 Dataset</b>	<b>11</b>
<b>3.3 Data Pre- Processing</b>	<b>13</b>
<b>3.4 Model Installation</b>	<b>13</b>
<b>3.5 Convolutional Neural Network (CNN)</b>	<b>13</b>
<b>3.6 Machine Learning Models</b>	<b>14</b>
<b>3.6.1 Resnet101v2</b>	<b>14</b>
<b>3.6.2 VGG16</b>	<b>15</b>
<b>Chapter 4: Results and Analysis</b>	<b>16</b>
<b>Chapter 5: Teamwork</b>	<b>19</b>
<b>Chapter 6: Conclusion&amp; Future work</b>	<b>20</b>
<b>References</b>	<b>21</b>

# Chapter 1: Introduction

---

In this project we will try to create vector from each and every image containing different types of objects and convert them into 1D array using different types of CNN models such as VGG16, RESNET etc. After creating vector, we will try to implement them in a graph and then using Euclidean distance or cosine similarity method we will try to find nearest 10 neighbors of each and every image and see what they represent for example if we give an image of dog, we will see what are the 10 closest thing that goes with the similar vector value and what are those pictures, are they similar to a dog picture or any other object. In this chapter, at first, we will explain about image to vector what is nearest neighbor and what techniques are there to detect them. The remaining part of the paper is organized as follows: in Chapter 2, a survey of the literature. The methodology is discussed in Chapter 3. Chapter 4 analyzes the results and their accuracy; finally, chapter 5 discusses some future work scopes and conclusions.

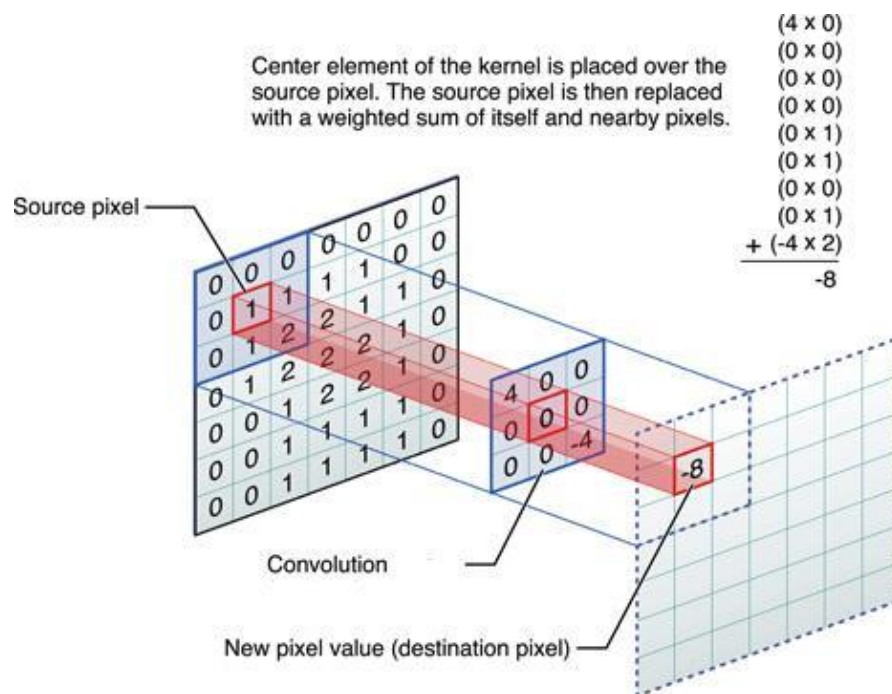
## 1.1 What is Image to Vector?

Convolutional neural networks, often known as CNNs or ConvNets, are a subset of deep neural networks used in deep learning. In numerous applications, including image and video identification, recommender systems, image classification, image segmentation, medical image analysis, and natural language processing, CNNs have demonstrated extraordinary state-of-the-art performance.

An example of a convolution occurs when a kernel, or tiny matrix of weights, is slid over input data and multiplies that portion of the input element-by-element before adding the results to the output.

Convolution is a mathematical operation on two functions ( $f$  and  $g$ ) in mathematics (specifically, functional analysis) that results in a third function ( $f * g$ ), which indicates how the form of one is changed by the other. A convolution, intuitively, enables weight sharing, which lowers the number of useful parameters, and picture translation (allowing for the same feature to be detected in different parts of the input space).

The 3D visualization of the convolution operation can be seen as follow,



Depending upon the type of the kernel, the different features from the input image can be extracted.

1 <small>x<sub>1</sub></small>	1 <small>x<sub>0</sub></small>	1 <small>x<sub>1</sub></small>	0	0
0 <small>x<sub>0</sub></small>	1 <small>x<sub>1</sub></small>	1 <small>x<sub>0</sub></small>	1	0
0 <small>x<sub>1</sub></small>	0 <small>x<sub>0</sub></small>	1 <small>x<sub>1</sub></small>	1	1
0	0	1	1	0
0	1	1	0	0

Image






4		

## Convolved Feature









## 1.2 What is Nearest neighbor of an image?

In the field of deep learning, neural networks are frequently used to learn vector representations of things. Then, we may use these vector representations to a wide range of practical activities.

Let's use the situation of a deep learning-based facial recognition system as a specific illustration. The aim in this use case is to determine whether or not a person in a provided photo matches a person in a database of known identities. The objects are photos of people's faces. To do face recognition, all that is required is to take the vector representation of a provided image (the query vector) and look for related vectors in our database. We will utilize a neural network to create vector representations of all of the photographs. In this definition, vectors that are near to one another in vector space are considered comparable.

	Known Identities		
Submitted Photo	 [0.209, 0.056, 0.033, ..., 0.160]	 [0.275, 0.268, 0.037, ..., 0.089]	 [0.050, 0.283, 0., ..., 0.017]
		 [0.276, 0.632, 0.020, ..., 0.782]	 [0.016, 2.930, 0.0197, ..., 0.270]

The closest neighbor's search method is used to locate vectors that are close to our query. Calculating the distance between the query vector and each vector in our collection is a crude way to perform closest neighbor's search (commonly referred to as the reference set). However, when your reference collection expands to millions of items, computing these distances by sheer force rapidly becomes impossible.

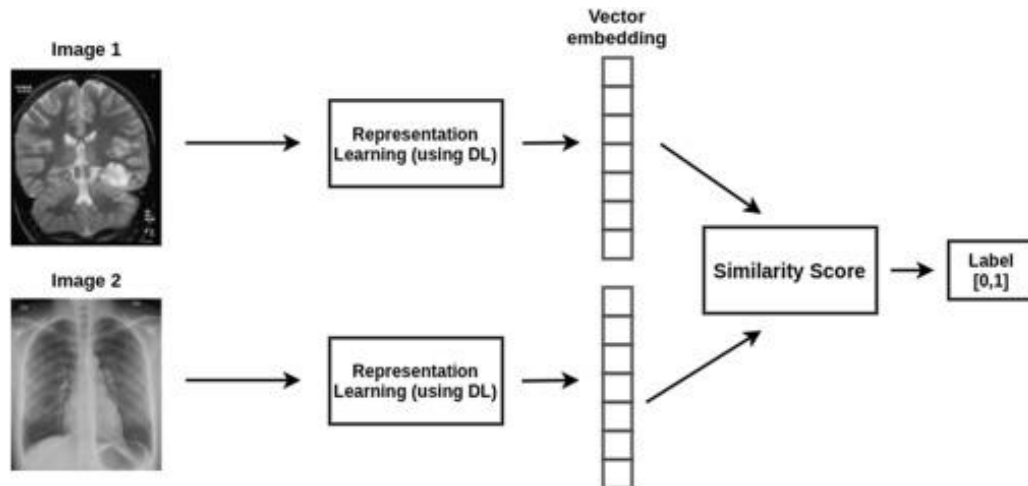
Calculated Distances (brute force)		Queries		
				
Reference Set	1 	13.2	32.7	4.8
	2 	22.4	21.4	9.7
	3 	5.9	8.1	21.2
	...	...	...	...
	N-1 	36.7	22.9	26.4
	N 	29.8	34.7	31.8

### 1.3 What is Euclidean distance?

The distance between two points is known as the Euclidean distance in mathematics. In other words, the length of the line segment between two points is what is meant by defining the Euclidean distance between two locations in Euclidean space. It is also referred to as the Pythagorean distance since the Euclidean distance may be calculated using coordinate points and the Pythagoras theorem.

## 1.4 Cosine similarity

Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis.



## 1.5 Motivations

As there is not so much work done in this aspect and this is interesting work so we wanted to explore more about this and dug deep in this matter and observe what are the reasons of different images being represent of an image rather than representing all similar to that object.

## 1.6 Aims and Objectives

The purpose of this work is to observe and increase the accuracy of the similarity score of each and every object nearest neighbors as much as possible so that after it done, we can observe and do other researches of the reason and ways behind it.

# Chapter 2: Literature Review

---

A literature review is a summary of previous research on a particular topic. Reviewing existing research in a particular field of study is the goal of a literature review. Learning about our field of study which is image recognition is enhanced by conducting a literature review. However, we're looking forward to read three machine learning based approaches that's relevant to our project. This knowledge will be presented in a written report in this chapter.

## 2.1 Related Paper-1

Firstly, a paper named “Image reconstruction from ResNet semantic feature vector” was read. This paper was published by Vít Líst'ík, Dept. of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague. Vít Líst'ík et al [1] want to prove that although it is possible to reconstruct the image from the semantic feature vector. The task is to generate an image from the semantic feature vector which will be very similar to the original image. The task is the same as for autoencoder with a difference of using pre-trained CNN. They used only the images, not the labels. The original dataset consists of 14M labeled images. They are using random subsets of the dataset. They used pre-trained ResNet for the extraction. Based on their results they concluded that it is not possible to reconstruct the private information. Using this method, they were able to accurately identify 95.2 percent. Rather than using reconstruct the image from the semantic feature vector cosine similarity was our choice for this project.

## 2.2 Related Paper-2

In the second paper named “Local Aggregation for Unsupervised Learning of Visual Embeddings”, Chengxu Zhuang et al. [2] used a neural network or Local Aggregation (LA) method which nonlinearly embed inputs in a smaller space and they identified close neighbors and background neighbors. They identified two sets of neighbors for an  $x_i$  and its embedding  $v_i$ . They used  $B_i$  for Nearest-neighbor based identification and  $C_i$  for Robustified clustering-based identification, to identify close neighbors, they applied an unsupervised clustering algorithm. They followed the methods of AlexNet and VGG16 architectures, to add batch normalization (BN) layers in their experiment. They used K-nearest neighbor (KNN) classification results using the embedding features.



With all methods, Local Aggregation (LA) performs much better than alternative methods. LA trained ResNet-50 achieves 60.2% top-1 accuracy on ImageNet classification. For the LA method, they consistently see performance gains from both overall deeper structures and from early layers to deeper layers within an architecture. Rather than using AlexNet we use resnet101v2 for our project.

## **2.3 Related Paper-3**

In the third paper named “NEAREST NEIGHBOUR STRATEGIES FOR IMAGE UNDERSTANDING”, This paper was published by Sameer Singh students of University of Exeter Exeter EX4 4PT, UK. They use Nearest neighbor methods provide an important data classification tool for recognizing object classes in pattern recognition domains. Their main objective of this paper is to develop two versions of the nearest neighbour method. First model will resolve conflicts in the k-nearest neighbour rule, second is closest average distance of samples of classes involved. They described traditional nearest neighbour rule for their recognition. The results shown in their paper for our two nearest neighbour models are extremely encouraging.

## Chapter 3: Methodology

---

This chapter gives an overview of the different parts of the work chronologically. It mainly discusses the work's theories, techniques, and step-by-step workflow. To complete this part of our project, we have used Resnet101v2 and vgg16 models which is an open-source machine learning model.

### 3.1 Workflow

The figure depicts the suggested method's complete process diagram.

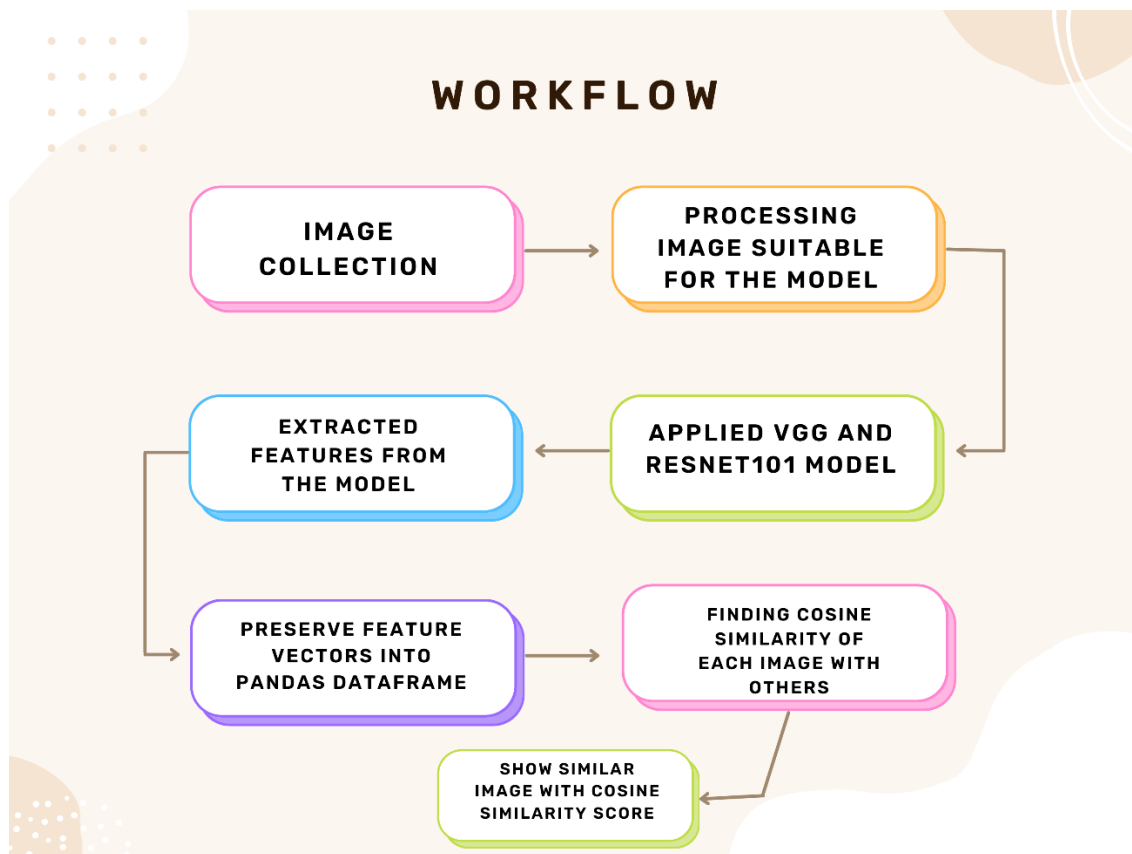


Figure 1: Nearest Neighbour of Different Image Representation.

The workflow describes the whole process of Nearest Neighbour of Different Image Representation. After giving the input image, the output will be shown 10 Nearest Neighbour for this image with an estimated accuracy.

## 3.2 Dataset

For this project we are going to use the ImageNet dataset. But now we used our own dataset. The dataset into 10 classes and Every class contains 100 images. Total 1000 image we use for our project in this course. The dataset contains car, bird, dog, cat and other varieties picture , which was not efficient enough so we want to gradually shift the dataset to the ImageNet dataset as soon as possible. Here is the link of our dataset. <https://www.kaggle.com/competitions/imagenet-object-localization-challenge/data?select=ILSVRC>

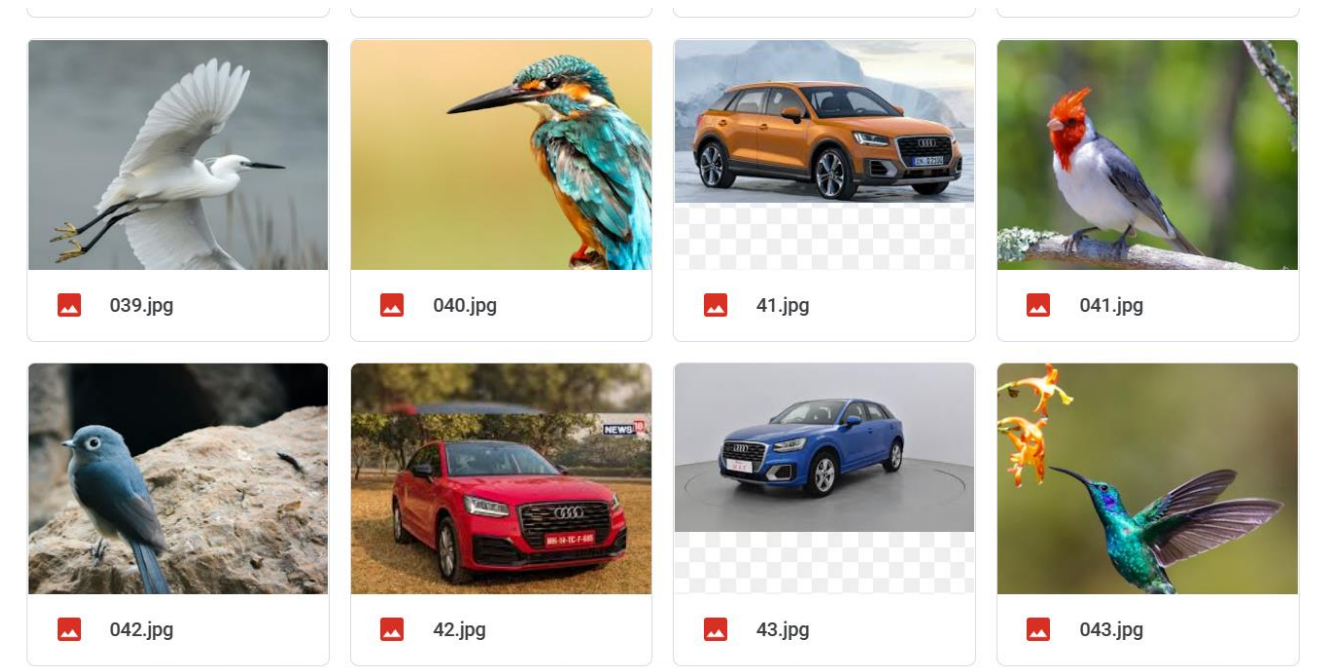
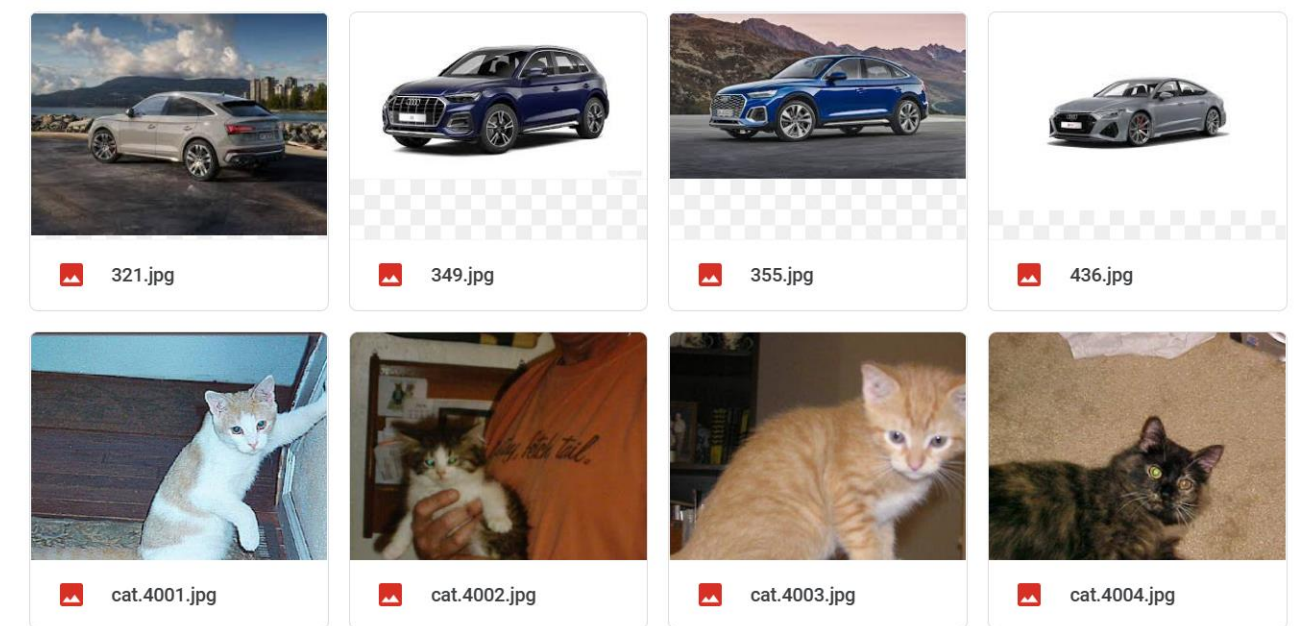
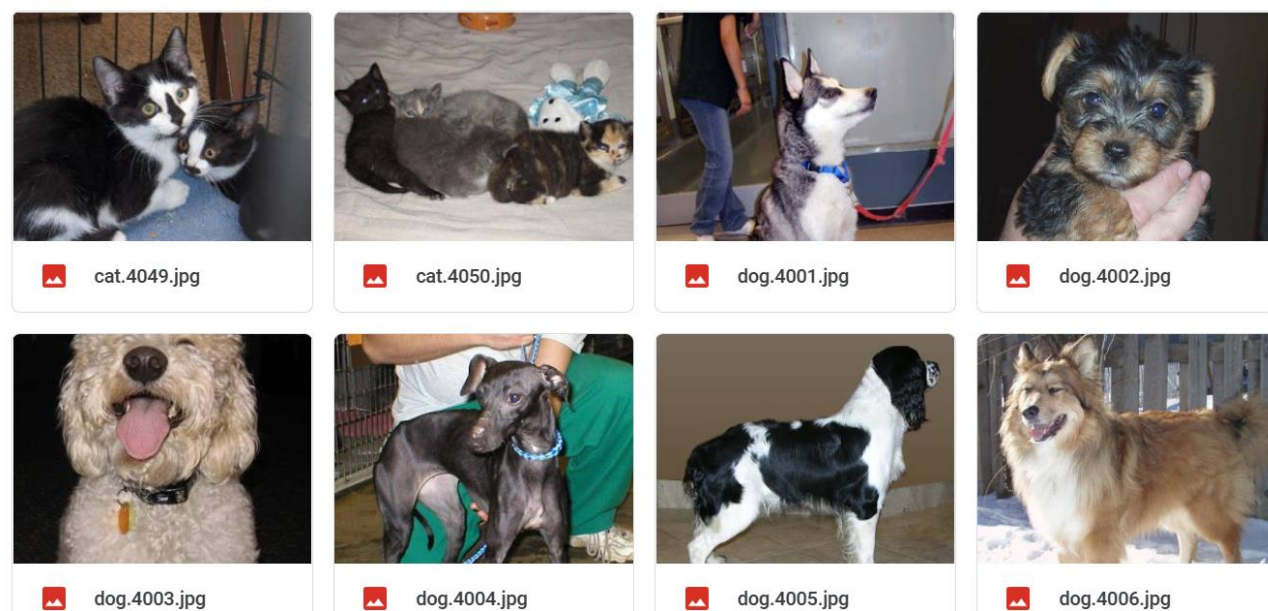


Figure2: Own dataset



**Figure3: Own dataset**



**Figure4: Own dataset**

### **3.3 Data Pre- Processing**

Pre-processing images is a critical stage in enhancing the effect of picture classification. Because the CNN learning method coordinates the execution of our machine learning activity, we classified and resized the images for training and testing during the image pre-processing step.

### **3.4 Model Installation**

To begin, we use google colab. Following that, we downloaded the Resnet101v2 and vgg16. Additionally, we employed the transfer learning method, which retains the parameters from the previous layer, to eliminate the final layer of the Resnet101v2 model and retrain a new layer.

### **3.5 Convolutional Neural Network (CNN)**

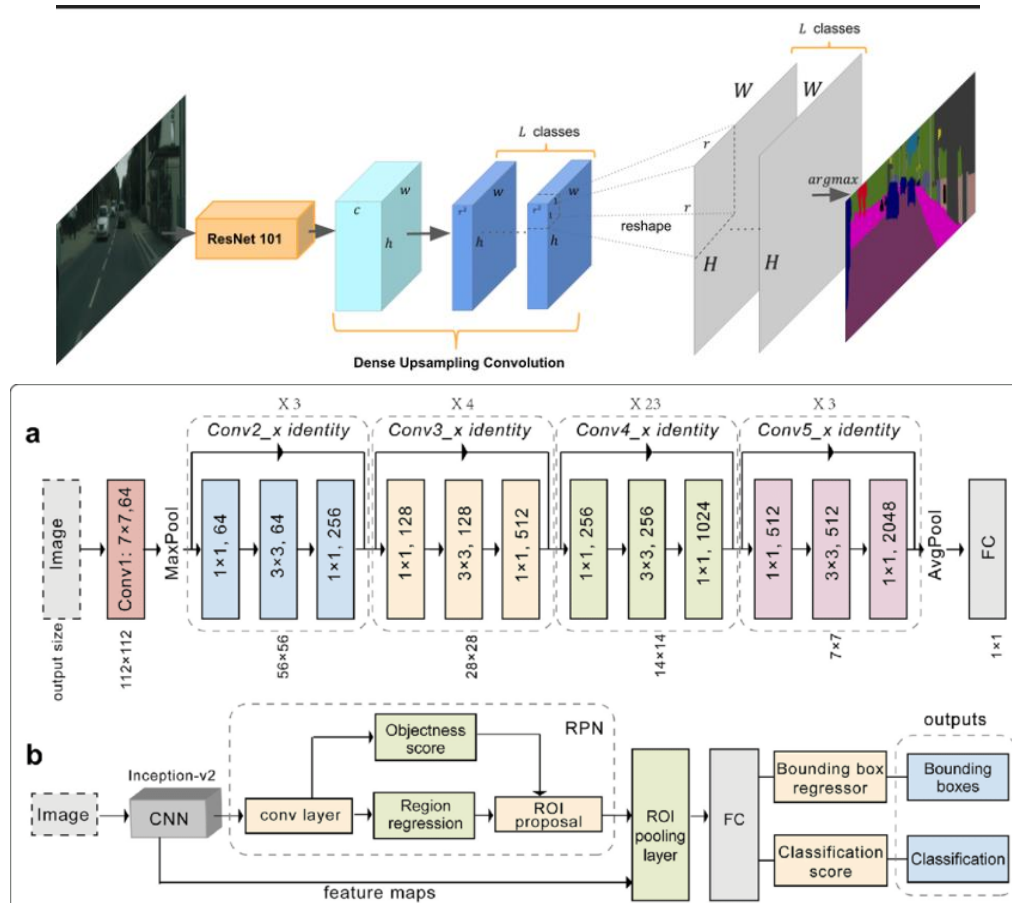
CNNs are multi-layer artificial neural networks capable of both unsupervised and supervised feature extraction and classification. A CNN is composed of a series of convolutional and pooling layers used to extract features, followed by one or more fully connected layers used for classification. Sparse connectivity and weight sharing are characteristics of convolutional layers. A convolutional layer's units receive their inputs from a small rectangular subset of the previous layer's units. Additionally, the nodes of a convolutional layer are grouped in weighted feature maps. Each feature map's inputs are tiled to correspond to overlapping regions of the previous layer, equating the aforementioned procedure to convolution, while the shared weights within each map correspond to the kernels. Convolution produces nonlinearities at the element level when the output is passed through an activation function. Following this is a pooling layer that subsamples the previous layer by aggregating small rectangular subsets of values. Max or mean pooling is used to substitute the maximum or mean value for the input values. Following that are a series of fully connected layers, the final one having a unit count equal to the class count. This section of the network performs supervised classification and receives as input the values from the previous pooling layer that comprise the feature set. The CNN is trained using back propagation and the gradient descent method.

## 3.6 Machine Learning Models

Given that this is a classification experiment, we selected well-known classifiers that are appropriate for our project.

### 3.6.1 Resnet101v2

ResNet-101 is a convolutional neural network that is 101 layers deep. we can load a pretrained version of the network trained on more than a million images from the ImageNet dataset. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.



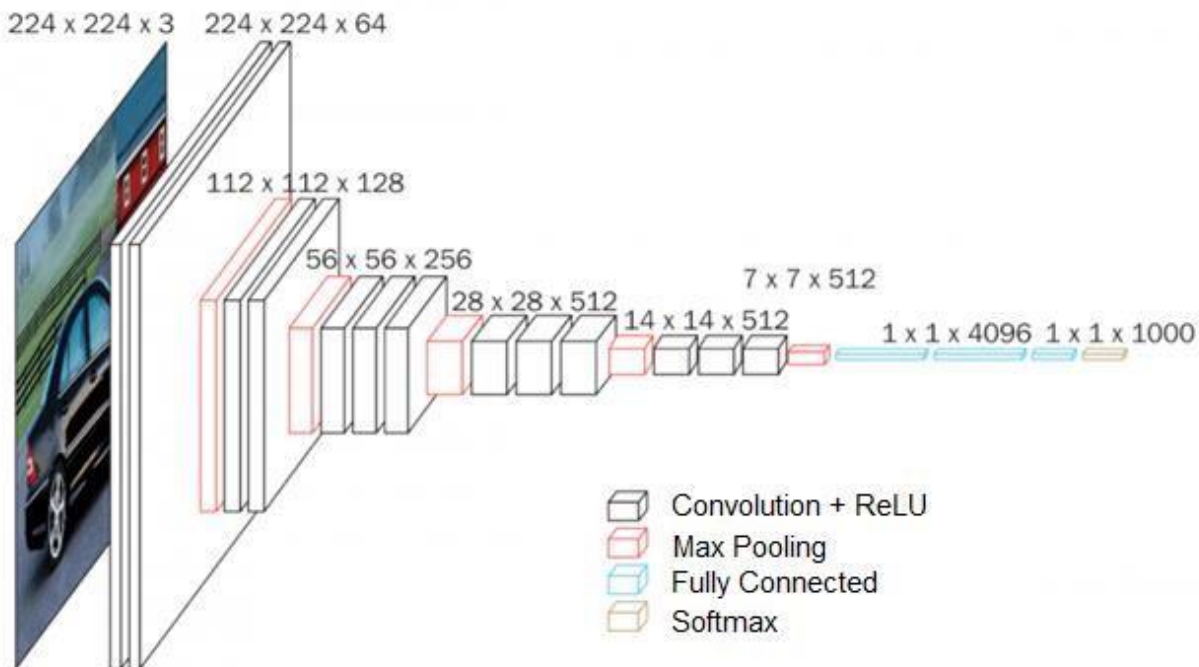
**Figure 5: Resnet101 Architecture Diagram.**

Using the dataset as a training set, the machine can detect 10 nearest neighbours for this image.



### 3.6.2 VGG16

VGG16 is a convolutional neural network (CNN) architecture that is widely regarded as one of the best available vision model architectures. Rather than having a large number of hyperparameters, VGG16 focused on 3x3 convolution layers with stride 1 and always used the same padding and maxpool layer of a 2x2 filter with stride 2. Throughout the architecture, it maintains this arrangement of convolution and max pool layers. Finally, the output is handled by two FC and a softmax. The 16 in VGG16 refers to the sixteen weighted layers contained within. This is a truly massive network with over 138 million parameters.



**Figure 5: VGG16 Architecture Diagram.**

Any of the network setups accepts an input image with a fixed size of 224 by 224 pixels and three channels – R, G, and B. The only pre-processing is for each pixel's RGB values to be normalized. This is accomplished by deducting the average value from each pixel. The input to any of the network configurations is considered to be a fixed size 224 x 224 image with three channels – R, G, and B. The only pre-processing done is normalizing the RGB values for every pixel. This is achieved by subtracting the mean value from every pixel.

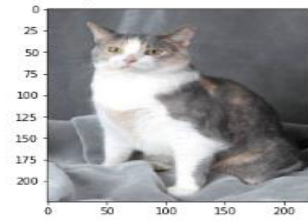
## Chapter 4: Results and Analysis

In this research, we investigate how deep learning methods may be used to create a feature extractor model and obtain the most comparable photos from a dataset contains 10 classes and every class contains 100 images. SO total 1000 image we use for our project. ResNet-101 and Vgg16 are pre-trained models from the Keras library that were employed in this study. To extract features from the provided dataset, a feature extractor model was constructed based on the pre-trained model. A function to determine the cosine similarity between the photos was also suggested. 1000 photos representing 10 distinct categories may be found in the collection. The sizes of the photos varied. After loading the images into a directory, a numpy array was created from them. The images were pre-processed by leveling the pixel values and making them all the same size. In order to work with the pre-trained model. The input image is 224x224x3 pixels in size. The input image is 224x224 pixels wide and tall, and the number 3 stands for the number of red, green, and blue channels in the image (RGB). The input data is changed from a two-dimensional array to a one-dimensional array using the flatten layer of the code. The flatten layer of this code is used to convert the input data from a two-dimensional array to a one-dimensional array. The flatten layer is also used to reduce the complexity of the model and improve its speed. The feature extractor model was built on the pre-trained model to extract features from the given dataset. The model was implemented using the Keras library. The pre-trained models used in this study were Vgg16 and ResNet-101. The features were then extracted from the images and stored in a feature vector. The cosine similarity between the images was calculated using the dot product and the magnitude of the feature vector. The cosine similarity score was used to determine the similarity between the images. Here, In the output section of ResNet 101 the original image was cat, and from the 10 nearest neighbors, we got 5 similar photos of cat with similarity score out of 10. We followed a similar procedure for Vgg16 and also got 5 similar images of cat and 5 different categories. Here is the output for our project:

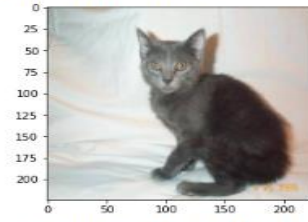


```
[ ] retrieve_most_similar_products(files[40])
```

original product:



most similar products:

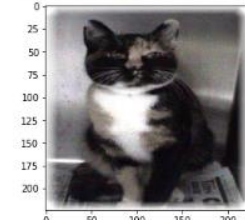


similarity score : 0.76413625

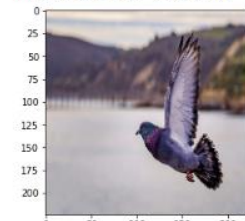


similarity score : 0.75220734

similarity score : 0.7130859



similarity score : 0.71262425

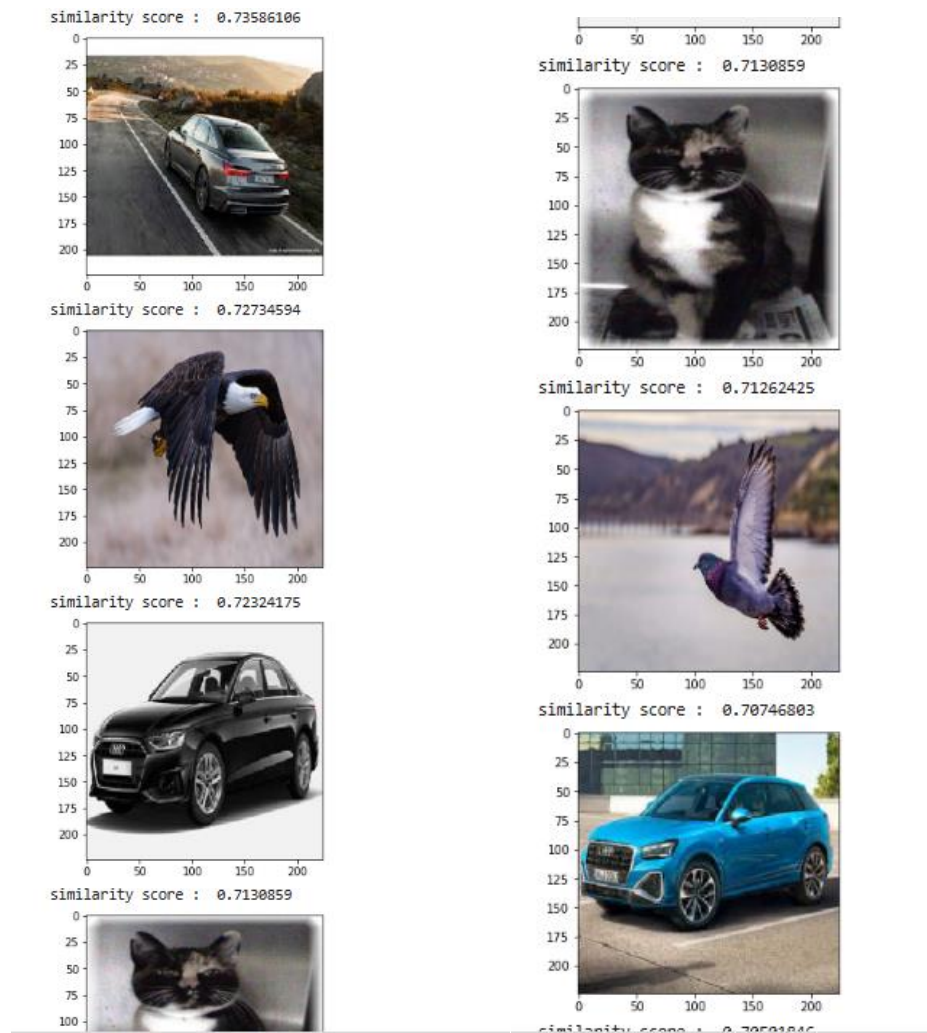


similarity score : 0.70746803



similarity score : 0.70591816

**Figure: Output 10 nearest neighbours**



**Figure: Output 10 nearest neighbours**

## Chapter 5: Teamwork

- From the beginning of this course, we started our teamwork from project topic selection and the approaches we should follow to complete the project.
- During the model training we have faced some problems which we solved together. To bring more accuracy we modified the model.
- We always divided work when we needed and got together when we couldn't solve something or come to a conclusion. When one of us couldn't attend do perform any work other member made sure he covered that for him
- In this very way, we have maintained our teamwork and completed a part of the project successfully.
- We had to utilize the VGG16 model alongside with Resnet101V2 model in order to achieve higher similarity of 82%.

## **Chapter 6: Conclusion& Future work**

We have thus far completed the job with the anticipated results after overcoming a few challenges. We look into how a feature extractor model may be built using deep learning techniques to find the most similar images from a dataset of 1000 images across 10 classes. After that, the photos' features were taken out and put in a feature vector for storage. We want to use some new techniques in the future to further our work.

# References

- [1] [http://poseidon2.feld.cvut.cz/conf/poster/poster2018/proceedings/Poster\\_2018/Section\\_IC/IC\\_057\\_Listik.pdf](http://poseidon2.feld.cvut.cz/conf/poster/poster2018/proceedings/Poster_2018/Section_IC/IC_057_Listik.pdf)
- [2] [Local Aggregation for Unsupervised Learning of Visual... - Google Scholar](#)
- [3] [NEAREST NEIGHBOUR STRATEGIES FOR IMAGE UNDERSTANDING \(psu.edu\)](#)N. Barla, "v7labs," v7, 15 March 2022. [Online]. Available: <https://www.v7labs.com/blog/semantic-segmentationguide#:~:text=Semantic%20Segmentation%20follows%20three%20steps,b%20creating%20a%20segmentation%20mask>. [Accessed 24 April 2022].
- [4] KHLOE, "Image Recognition Applications," 16 January 2022. [Online]. Available: <https://www.datasciencesociety.net/image-recognition-applications-7-essential-future-uses/>. [Accessed 25 April 2022].
- [5] R. I. S. B. S. Nishat Tasnim, "A Convolution Neural Network Based Classification Approach
- [6] for Recognizing Traditional Foods of Bangladesh from Food Images," *A Convolution Neural Network Based Classification Approach for Recognizing Traditional Foods of Bangladesh from Food Images*, p. 4, 2020.
- [7] Pawangfg, "Residual-networks-resnet-deep-learning," geeksforgeeks, 27 January 2022.
- [8] [Online]. Available: <https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/>. [Accessed 22 April 2022].
- [9] A. Thite, "Introduction to VGG16," Great Learning, 1 October 2021. [Online]. Available:
- [10] <https://www.mygreatlearning.com/blog/introduction-to-vgg16/>. [Accessed 23 April 2022].
- [11] [On Vectorization of Convolution Layer in Convolution Neural Networks \(CNNs\) | by Sanghvirajit | Analytics Vidhya | Medium](#)
- [12] [Scaling nearest neighbors search with approximate methods. \(jeremyjordan.me\)](#)