**CIHE22420 | KAMINI JOSHI**

**Assessment Brief- Assessment 2**

**Assessment Title: Project Development and Analysis in Emerging Technologies**

Table of Contents

[Abstract 3](#_Toc175067406)

[Introduction 3](#_Toc175067407)

[Methodology 5](#_Toc175067408)

[Proposed Method for image classification 5](#_Toc175067409)

[Convolutional Neural Network Structure 6](#_Toc175067410)

[ Input Layer: 6](#_Toc175067411)

[ Convolutional Layers: 6](#_Toc175067412)

[ Activation Functions: 6](#_Toc175067413)

[ Pooling Layers: 6](#_Toc175067414)

[ Fully connected Layers: 6](#_Toc175067415)

[ Output Layer: 7](#_Toc175067416)

[Dataset Description 7](#_Toc175067417)

[Tools Required 8](#_Toc175067418)

[Google colab: 8](#_Toc175067419)

[Tensorflow and Keras: 8](#_Toc175067420)

[ Tesnorflow: 8](#_Toc175067421)

[ Keras: 8](#_Toc175067422)

[Implementation 9](#_Toc175067423)

[Import Libraries 9](#_Toc175067424)

[Load dataset 9](#_Toc175067425)

[Preprocessing of dataset 10](#_Toc175067426)

[Build CNN Model 11](#_Toc175067427)

[Model Compiling and Training 13](#_Toc175067428)

[Results Analysis 14](#_Toc175067429)

[Conclusion 16](#_Toc175067430)

[References 17](#_Toc175067431)

# Abstract

Due to its simplicity, image classification has gained appeal as a fundamental application of deep learning. The Convolutional Neural Network (CNN) is a widely used data architecture for image classification. CNN is supervised machine learning technology. It is primarily used for image processing and voice recognition. Convolutional Neural Networks (CNNs) are currently at the forefront of image classification. Below, we present a comprehensive summary of the many components of CNN. This project provides a detailed explanation of the process of classifying images using convolutional neural networks (CNNs). We have discussed the intricacies of training the cnn model and its rationale for classifying images. The cats and dogs dataset is used for this project. We endeavored to create a CNN-based image classifier with the ability to differentiate between cats and dogs. Our model has exceptional accuracy in image classification, as evidenced by the 84% accuracy achieved in the experimental phase. This high accuracy was attained by training the images using Sequential CNN networks.

# Introduction

The fast growth in the field of Artificial intelligence image recognition and classification is also a major challenge for visual information indexing and retrieval systems. Computer vision, an interdisciplinary branch of artificial intelligence, tries to help computers understand images like humans. Multiple research projects have addressed these difficulties.

Visual interpretation and classification are easy for humans but difficult for computers. A typical image has pixels with different values. Image classification requires more calculations. This requires systems with higher specs and computational capacity.

In 2016 authors (Chen, Jiang, Li, Jia, & Ghamisi, 2016) studied deep learning and feature extraction from Hyper Spectral Images. Convolutional Neural Networks with several pooling layers can extract nonlinear and invariant properties from the Histogram of Orientated Gradients (HOG). These properties improve image classification and target detection.

Over the past decade, many image classification approaches have been reported and compared. Image classification involves labeling pixels into classes to extract data. This can be done with supervised or unsupervised classification. In 2016, writers (Chuang, Hwang, & Williams, 2016) addressed the topic of using an unsupervised learning algorithm for image classification in an underwater fish recognition framework.

Currently, deep learning algorithms are yielding favourable outcomes in fields such as computer vision. The image classification task is now employing the Convolutional Neural Network, which is a machine learning technique. The authors(Kim et al., 2016) said that a deep learning method is employed to classify the grade of wood boards based on the retrieved texture information from the wood photos. In addition, he drew a parallel between the architecture of machine learning. CNN, short for Convolutional Neural Network, is a specific sort of artificial neural network that has proven to be effective at analysing visual images. The design is influenced by biological processes, with the neurones interconnected in a manner similar to that of the visual cortex in animals. The author(Silva, Welfer, Gioda, & Dornelles, 2017) has presented a discussion on the utilisation of Convolutional Neural Networks (CNN) for the automatic recognition of cattle images. This approach aids in extracting the essential features from the images, and the classification of these images is performed using Support Vector Machine (SVM) approaches.

Classification uses a deep convolutional neural network. CNN has input, hidden, and output layers. Pixel matrices are typical image displays. Pixel values, weights, and biases (for non-linearity) are entered into the input layer. A fully connected layer is used to identify images in the output layer. Convolutional layer are hidden layer used for features finding. The article in reference (Lunga, Prasad, Crawford, & Ersoy, 2013) explores various manifold learning approaches used for the classification of distantly sensed hyperspectral data.

The basic Convolutional layer has adjustable filters and settings. Each filter is narrow in width and height but covers the whole input volume's depth. The dot product between the input and filter entries for each filter produces the 2-dimensional activation map. The network learns about filters that activate when they identify specified properties in specific input places. The pooling layer reduces image size while keeping information. Max pooling selects the highest value from preceding layer neurones. Full connectivity creates connections between all neurons in one layer and another. CNNs require less pre-processing than manual filter-based categorization algorithms. CNN learns filters without human intervention, a major benefit.Convolutional Neural Networks (CNNs) are deep learning algorithms that require supervision and a lot of labelled data to train. The model will learn the weights after training, improving classifier accuracy.

# Methodology

## Proposed Method for image classification

Convolutional neural networks are utilised in the interdisciplinary field of deep learning for image identification and classification. The classification of images will be accomplished by the utilisation of convolutional neural networks with the assistance of human involvement. The human utilises the realtime images databases, in our case images of cats and dogs.

The input dataset containing images of cats and dogs At first, the human will instruct the classifier to extract the target pattern from the images. The images are thereafter classified using the pattern that was accurately derived from the preceding steps. The expertise of the classifier has a determining factor in the final output, which could differ from one set of patterns to another. (Hinton & Salakhutdinov, 2006; Krizhevsky, Sutskever, & Hinton, 2012) The deep learning architecture for image classification has been studied, which involves the use of several layers in a Convolutional Neural Network (CNN) to extract novel features from the images datasets. The project endeavours to tackle the issue of image classification by utilising a rudimentary convolutional neural network. The dataset used in this study comprises of cat and dog images sourced from Kaggle.

## Convolutional Neural Network Structure

The writers (Sharma, Jain, & Mishra, 2018) have discussed that one subset of deep neural networks, Convolutional Neural Networks (CNNs) are trained to process and understand data presented in a grid format, such as imges. They have multiple layers, and each one is responsible for something specific. Parts that are absolutely necessary for a CNN to function include are given below.

### Input Layer:

The input layer of a neural network receives the unprocessed pixel values of an image and acts as the initial stage of the network.

### Convolutional Layers:

Convolutional layers use kernels to filter input images. Each filter learns to recognize edges, corners, and textures. Convolutional layers build feature maps by filtering the input image and multiplying and summing elements.

### Activation Functions:

Introducing non-linearities with activation functions helps the network learn complicated associations. ReLU, sigmoid, and hyperbolic tangent are common activation functions for CNNs.

### Pooling Layers:

Pooling layers downsample convolutional layer feature maps to minimize their size. Max pooling chooses the greatest value in each pooling zone, reducing complexity while keeping key properties.

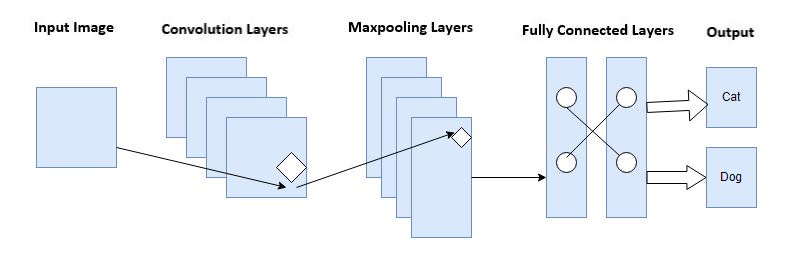
### Fully connected Layers:

Fully connected layers, sometimes called dense layers, connect every neuron in the previous layer to the next. Thus, the network may obtain sophisticated input data representations. Fully linked layers usually end with a final activation function like softmax that generates class probabilities.

### Output Layer:

The output layer generates the final predictions or classifications using the acquired representations.

The figure 1 explain the components of CNN networks.



**Figure 1: CNN Architecture Diagram for Image Classification**

# Dataset Description

The dataset titled "Cats and Dogs" is utilised in this research. It is obtained by from the Kaggle (https://www.kaggle.com/datasets/tongpython/cat-and-dog). It contains two files training\_set and test\_set. Additionally, the set includes two folders with the names cat and dogs. 10028 pictures are included in the dataset. 8005 photographs of cats and dogs are included in the training folder, while 1023 photographs of cats and dogs are included in the test folder. Diagrams 2 and 3 are examples of images from the collection.



**Figure 2: sample image of cat in dataset Figure 3: sample image of dog in dataset**

# Tools Required

The following technologies and tools are utilised in the completion of this project.

## Google colab:

Google Colab GPU is used for model training and classification. Colab is particularly well suited for machine learning, deep learning projects.

## Tensorflow and Keras:

Tensorflow, Keras python libraries are utilised for dataset preprocessing and model building.

### Tesnorflow:

TensorFlow is an open-source software framework developed by Google in late 2015. It is specifically developed to facilitate the construction of a diverse array of machine learning and deep learning models. The main objective of using TensorFlow is to reduce the complexity involved in doing computations on large numerical datasets. TensorFlow receives inputs in the form of a multi-dimensional array referred to as a Tensor. The three primary components of TensorFlow are data preprocessing, model construction, and model training and estimation(Vasilev, Slater, Spacagna, Roelants, & Zocca, 2019).

### Keras:

Using human readability and structures that are basic and consistent, the application programming interface (API) known as Keras, which is built on Python. Keras is commonly referred to as a wrapper for TensorFlow by some individuals. The purpose of this is to facilitate fast experimentation(Vasilev et al., 2019).

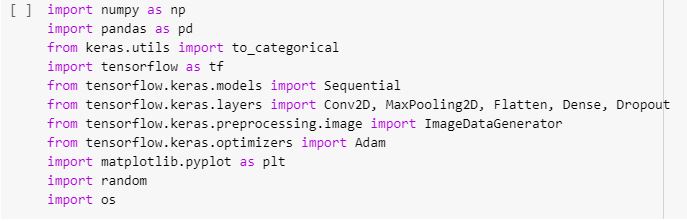
For both the visualisation of datasets and the plotting of graphs, the Matplotlib package is utilised.

# Implementation

Following steps are performed for Implementation of Image classification using convolution neural network.

## Import Libraries

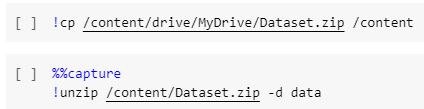
It is important to import important libraries like as TensorFlow and MatplotLib. For the purpose of importing necessary libraries, the following snippet of code is utilised.



## Load dataset

Zip file containing a dataset folder. To begin, mount Google Drive on google colab jupyter notebook . The dataset should be unzipped once Google Drive has been mounted. Next, after the dataset file has been unzipped, the path for the training and testing folders should be defined. All of this process is carried out using the following snippet of code.







## Preprocessing of dataset

After defining the training and testing folder path. Next, prepare the training and testing dataset. Data augmentation expands the training dataset. Prepare training and testing datasets using rescaling and augmentation. The training dataset is rescaled and augmented. Additionally, only the test dataset is rescaled. The training dataset has 8005 images for both classes, whereas the test dataset has 2023. This operation is completed with the following code.

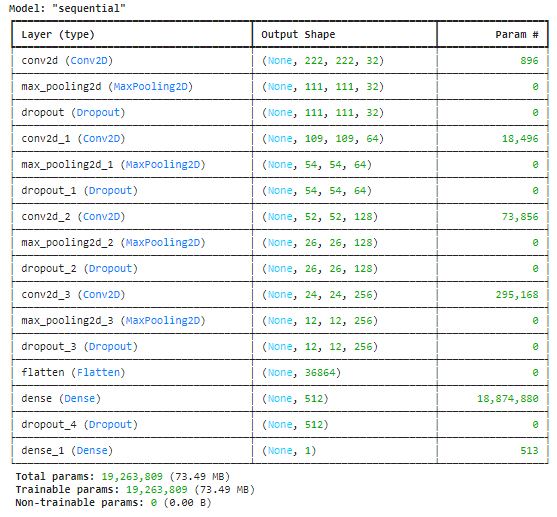


The following snippet of code is used to display one image from each of the classes. Figure 2 and 3 are output of this snippet of code.



## Build CNN Model

A basic convolutional network is a series of layers. Our network design is constructed using five distinct layers convolutional layer, pooling layer, dropout layer, flatten layer, and fully connected layer. Figure 4 depicts the structure of our proposed Convolutional Neural Network (CNN) model. The initial layer is a convolutional layer with size of 3x3. The activation function used is relu, and the input image has dimensions of 224x224x3. Next, incorporate a maxpooling layer with a size of 2x2, followed by the addition of a dropout layer. Following this, three further convolution layers, maxpooling layers, four dropout layers, one flatten layer, and two dense layers were included.



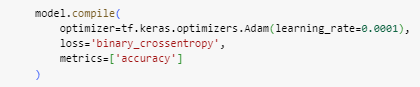
**Figure 4: Proposed CNN model summary**

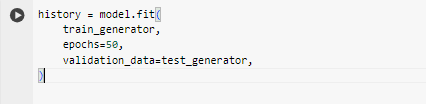
And the part of the code for the model is.

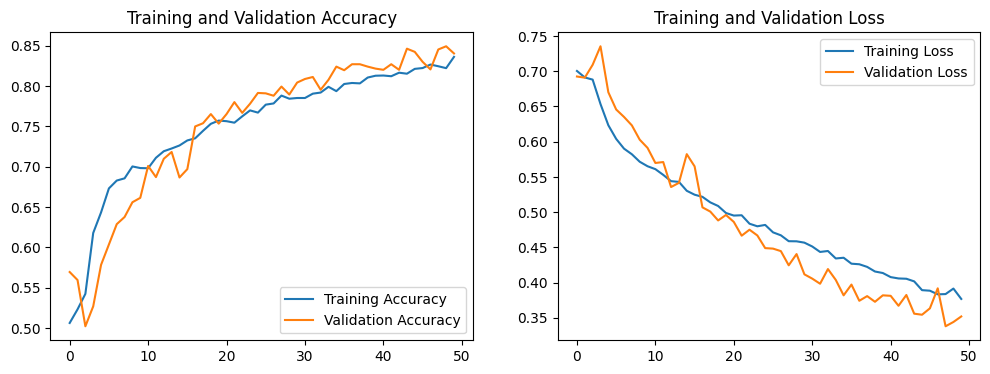


## Model Compiling and Training

Now initialize parameters epochs, batch size, and optimiser. Set batch size 32, epochs are 50 and Adam optimiser learns at 0.0001. Supplying the validation dataset to validate the model at each epoch during training. Training accuracy, loss, validation accuracy and loss are shown for each epoch in the training history. Training and compilation code is below. Figure 5 contains graphs for training and validation accuracies and losses.



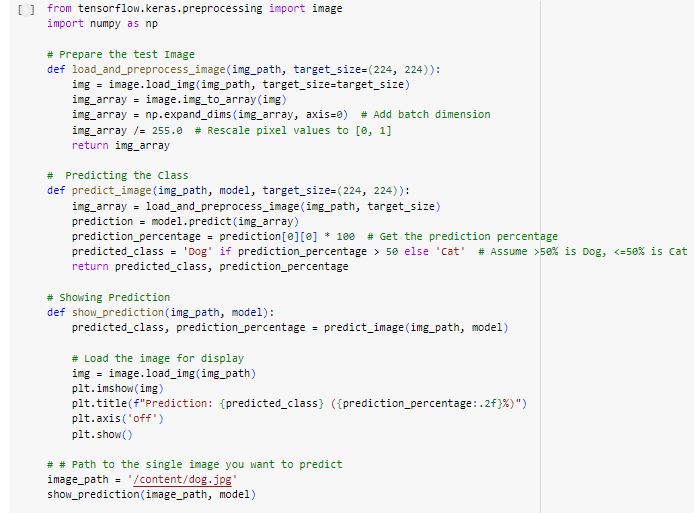




**Figure 5: graph for training and validation accuracy and Loss**

# Results Analysis

The model is trained across 50 epochs. Here, optimiser Adam is used. Initially, we trained with 10 epcohs and the Adam optimizer with no dropout layer. When training began, the model overfitted and did not perform as planned. We add one dropout layer and increase epochs from 10 to thirty. We also boost Adam optimiser learning rate to 0.001. After updating the parameter, the model was retrained. However, this strategy improves model training and validation accuracy. However, validation loss increased after 15 epochs. We then increase the learning rate from 0.001 to 0.0001 and train the model on 50 experiences. Additional dropout layers are added. These model training changes boosted validation accuracy and decreased validation loss. Due to fifty epcohs, practice time has risen. After 50 epochs, we achieved 84% training accuracy and 84.15% validation accuracy. We can additionally boost it by training the model for 100 epochs and increasing the training dataset. After training, the model had 84% accuracy on the test dataset. Additionally, we test our model on one single image. Following this image downloaded from google, we will transmit it to our model for a forecast. Predict it accurately with the model. Please see the code below for the model prediction on single.





**Figure 6: CNN Model correctly prediction**

# Conclusion

Sequential CNNs is used to classify images cats and dogs in this project. The suggested model's test dataset has 84% accuracy. Despite having small size of dataset in the training dataset, the model is still give accurate results on single image. The model is also evaluated for single-image prediction. The computing time to process these images for 50 epochs is high. Using clusters of GPUs to deepen the model and by increasing the size of dataset to the results will improve of image classification. Increase the training dataset and by using pretrained CNN models can also improve accuracy.

# References

Chen, Y., Jiang, H., Li, C., Jia, X., & Ghamisi, P. (2016). Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE transactions on geoscience and remote sensing, 54*(10), 6232-6251.

Chuang, M.-C., Hwang, J.-N., & Williams, K. (2016). A feature learning and object recognition framework for underwater fish images. *IEEE Transactions on Image Processing, 25*(4), 1862-1872.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science, 313*(5786), 504-507.

<https://www.kaggle.com/datasets/tongpython/cat-and-dog>. Retrieved from: <https://www.kaggle.com/datasets/tongpython/cat-and-dog>

Kim, H., Koo, J., Kim, D., Jung, S., Shin, J.-U., Lee, S., & Myung, H. (2016). Image-based monitoring of jellyfish using deep learning architecture. *IEEE sensors journal, 16*(8), 2215-2216.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems, 25*.

Lunga, D., Prasad, S., Crawford, M. M., & Ersoy, O. (2013). Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning. *IEEE Signal Processing Magazine, 31*(1), 55-66.

Sharma, N., Jain, V., & Mishra, A. (2018). An analysis of convolutional neural networks for image classification. *Procedia computer science, 132*, 377-384.

Silva, C., Welfer, D., Gioda, F. P., & Dornelles, C. (2017). Cattle brand recognition using convolutional neural network and support vector machines. *IEEE Latin America Transactions, 15*(2), 310-316.

Vasilev, I., Slater, D., Spacagna, G., Roelants, P., & Zocca, V. (2019). *Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow*: Packt Publishing Ltd.