# LOGO INFRINGEMENT DETECTION USING CNN

*A Project Report submitted in the partial fulfillment of the*

*Requirements for the award of the degree*

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**


Submitted by

**SK.Karishma (19471A05I1)**

**Y.Lakshmi Priya (19471A05J5)**

**S.Supriya (19471A05H8)**

Under the esteemed guidance of

**Dr. K .Lakshminadh,M.Tech ,Ph.D.**

Professor




**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**NARASARAOPETA ENGINEERING COLLEGE**

**(AUTONOMOUS)**

**(Accredited by NAAC with A+ Grade and NBA under Cycle-1**
**NIRF rank in the band of 251-320 and ISO 9001:2015 Certified**
**Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada**
**KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET-522601)**


**2022-2023**

# NARASARAOPETA ENGINEERING COLLEGE
## (AUTONOMOUS)

## DEPARTMENT OF COMPUTER SCIENCE ANENGINEERING


## CERTIFICATE

**This is to certify that the project that is entitled with the name "**LOGO INFRINGEMENT DETECTIONUSING CNN**" is a bonafide work done by the team SK.KARISHMA(19471A05I1), Y.LAKSHMIPRIYA (19471A05J5), S.SUPRIYA (19471A05H8) in partial fulfilment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2022-2023.**

PROJECT GUIDE

**Dr. K .LakshminadhM.Tech ,Ph.D .**

**Professor**

PROJECT CO-ORDINATOR

**Dr. Sireesha .M, M.Tech., Ph.D**

**Assoc Professor**

HEAD OF THE DEPARTMENT

**Dr. S. N. TirumalaRao, M.Tech., Ph.D.,**

**Professor**

EXTERNAL EXAMINER

# ACKNOWLEDGEMENT

We wish to express my thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sri **M.V.KoteswaraRao,B.Sc.,**who took keen interest in us in every effort throughout this course. We owe out sincere gratitude to our beloved principal **Dr.M.Sreenivasa Kumar, M.Tech., Ph.D., MISTE., FIE(I).,** for showing his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude towards **Dr.S.N.Tirumala Rao, M.Tech.,Ph.D.,** HOD of CSE department and also to our guide***Dr.* K .Lakshminadh M.Tech., Ph.D** of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks towards **Dr. Sireesha.M, M.Tech., Ph.D** , Assoc.Professor & Project coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying out doubts which had really helped us in successfully completing our project.

By

SK KARISHMA (19471A05I1)
Y LAKSHPRIYA (19471A05J5)
S SUPRIYA (19471A05H8)

# Abstract

When logos are increasingly created, logo detection has gradually become a research hotspot across many domains and tasks. Recent advances in this area are dominated by deep learning-based solutions, where many datasets, learning strategies, network architectures, etc. have been employed. This project reviews the advance in applying deep learning techniques to logo detection. Firstly, we discuss a comprehensive account of public datasets designed to facilitate performance evaluation of logo detection algorithms, which tend to be more diverse, more challenging, and more reflective of real life. Next, we perform an in-depth analysis of the existing logo detection strategies and the strengths and weaknesses of each learning strategy. Subsequently, we summarize the applications of logo detection in various fields, from intelligent transportation and brand monitoring to copyright and trademark compliance. Finally, we analyze the potential challenges and present the future directions for the development of logo detection to complete this survey.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

# INSTITUTE VISION AND MISSION

## INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

## INSTITUTION MISSION

**M1**: Provide the best class infra-structure to explore the field of engineering and research

**M2**: Build a passionate and a determined team of faculty with student centric teaching , imbibing experiential, innovative skills

**M3**: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

## MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

**M1:** Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

**M2:** Impart high quality professional training to get expertize in modern software tools and technologies to cater to the real time requirements of the Industry.

**M3:** Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

# Program Specific Outcomes (PSO's)

**PSO1:** Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

**PSO2:** Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering .

**PSO3:** Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

# Program Outcomes

**1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**2. Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**9. Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**10. Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**11. Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**12. Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

## Project Course Outcomes (CO'S):

**CO425.1:**Analyse the System of Examinations and identify the problem.

**CO425.2:**Identify and classify the requirements.

**CO425.3:**Review the Related Literature

**CO425.4:**Design and Modularize the project

**CO425.5:**Construct, Integrate, Test and Implement the Project.

**CO425.6:**Prepare the project Documentation and present the Report using appropriate method.

## Course Outcomes – Program Outcomes mapping

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **C425.1** | | ✓ | | | | | | | | | | | ✓ | | |
| **C425.2** | ✓ | | ✓ | | ✓ | | | | | | | | ✓ | | |
| **C425.3** | | | | ✓ | | ✓ | ✓ | ✓ | | | | | ✓ | | |
| **C425.4** | | | ✓ | | | ✓ | ✓ | ✓ | | | | | ✓ | ✓ | |
| **C425.5** | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **C425.6** | | | | | | | | | ✓ | ✓ | ✓ | | ✓ | ✓ | |

## Course Outcomes – Program Outcome correlation

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **C425.1** | 2 | 3 | | | | | | | | | | | 2 | | |
| **C425.2** | | | 2 | | 3 | | | | | | | | 2 | | |
| **C425.3** | | | | 2 | | 2 | 3 | 3 | | | | | 2 | | |
| **C425.4** | | | 2 | | | 1 | 1 | 2 | | | | | 3 | 2 | |
| **C425.5** | | | | | 3 | 3 | 3 | 2 | 3 | 2 | 2 | 1 | 3 | 2 | 1 |
| **C425.6** | | | | | | | | | 3 | 2 | 1 | | 2 | 3 | |

**Note:** The values in the above table represent the level of correlation between CO's and PO's:

1. **Low level**
2. **Medium level**
3. **High level**

## Project mapping with various courses of Curriculum with Attained PO's:

| Name of the course from which principles are applied in this project | Description of the device | Attained PO |
|---|---|---|
| C3.2.4, C3.2.5 | Gathering the requirements and defining the problem, plan to develop a smart bottle for health care using sensors. | PO1, PO3 |
| CC4.2.5 | Each and every requirement is critically analyzed, the process model is identified and divided into five modules | PO2, PO3 |
| CC4.2.5 | Logical design is done by using the unified modelling language which involves individual team work | PO3, PO5, PO9 |
| CC4.2.5 | Each and every module is tested, integrated, and evaluated in our project | PO1, PO5 |
| CC4.2.5 | Documentation is done by all our four members in the form of a group | PO10 |
| CC4.2.5 | Each and every phase of the work in group is presented periodically | PO10, PO11 |
| CC4.2.5 | Implementation is done and the project will be handled by the hospital management and in future updates in our project can be done based on air bubbles occurring in liquid in saline. | PO4, PO7 |
| CC4.2.8 CC4.2. | The physical design includes hardware components like sensors, gsm module, software and Arduino. | PO5, PO6 |

# INDEX

# LIST OF FIGURES

# 1.INTRODUCTION

## 1.1 Introduction

The Expanding and massive production of visual data from companies and institutions, and the increasingly popularity of social system. Graphics logos are special class of visual objects extremely important to access the identity of something or someone. Logos are graphic productions that either recall some real world objects, or emphasize a name, or simply display some abstract signs that have strong perceptual appeal. Most of the research related to trademark recognition deals with the problem of content based indexing and retrieval in logo databases, with the goal of assisting the process of trademark registration. In this case the image acquisition and processing chain is controlled so that the images are of acceptable quality and are not distorted. A generic system for logo detection and recognition in images taken in real world environments must comply with contrasting requirements. On the one hand, invariance to a large range of geometric and photometric transformations is required to comply with all the possible conditions of image/video recording. Since in real world images logos are not captured in isolation, logo detection and recognition should also be robust to partial occlusions. At the same time, especially if we want to discover malicious tampering or retrieve logos with some local peculiarities, we must also require that the small differences in the local structures are captured in the local descriptor and are sufficiently distinguishing for recognition

## 1.2 Objective of the project:

This project deals with finding fake logo by matching and recognizing it with the original logo. This is done by dividing the image of logo into rows and columns and thus each cell has its index value. Taking the index value of each cell which belongs to the image of logo to be verified check it with the original image index value of the corresponding cell. From the relation between index values of both the original logo and the one being considered we could decide it whether it is fake logo or original logo. If the index values of all the cells are exactly matching with the actual logo then it is considered to be original logo, otherwise it is the fake logo. This process is achieved by using CNN logo detection and recognition algorithm.

# 2. LITERATURE SURVEY

## 2.1 Deep Learning

**Deep learning** (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.

**Deep Learning** architectures such as deep neural networks, deep belief networks, graph neural networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.

**Artificial neural networks** (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological brains. Specifically, neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analogue.

The adjective "**deep**" in deep learning refers to the **use of multiple layers in the network**. Early work showed that a linear perceptron cannot be a universal classifier, but that a network with a nonpolynomial activation function with one hidden layer of unbounded width can. Deep learning is a modern variation which is concerned with an unbounded number of layers of bounded size, which permits practical application and optimized implementation, while retaining theoretical universality under mild conditions. In deep learning the layers are also permitted to be heterogeneous and to deviate widely from biologically informed connectionist models, for the sake of efficiency, trainability and understandability, whence the "structured" part.

## 2.2Some Deep learning methods

1.**Feedforward neural network:**

- This type of neural network is the very basic neural network where the flow control occurs from the input layer and goes towards the output layer.
- These kinds of networks are only having single layers or only 1 hidden layer

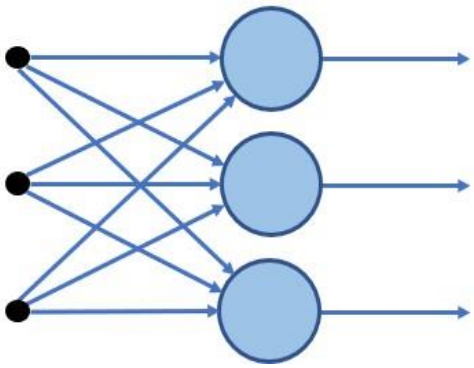


Fig:2.1 Feed Forward Neural Network

- In this network, the sum of the weights present in the input is fed into the input layer.
- These kinds of networks are used in the facial recognition algorithm using computer vision.

2. **Radial basis function neural networks:**
    - This kind of neural networks have generally more than 1 layer preferably two layers
    - In this kind of networks, the relative distance from any point to the center is calculated and the same is passed towards the next layer
    - Radial basis networks are generally used in the power restoration systems to restore the power in the shortest span of time to avoid the blackouts.
3. **Multi-Layer perceptron:**
    - This type of network are having more than 3 layers and its used to classify the data which is not linear
    - These kinds of networks are fully connected with every node.
    - These networks are extensively used for speech recognition and other machine learning technologies.

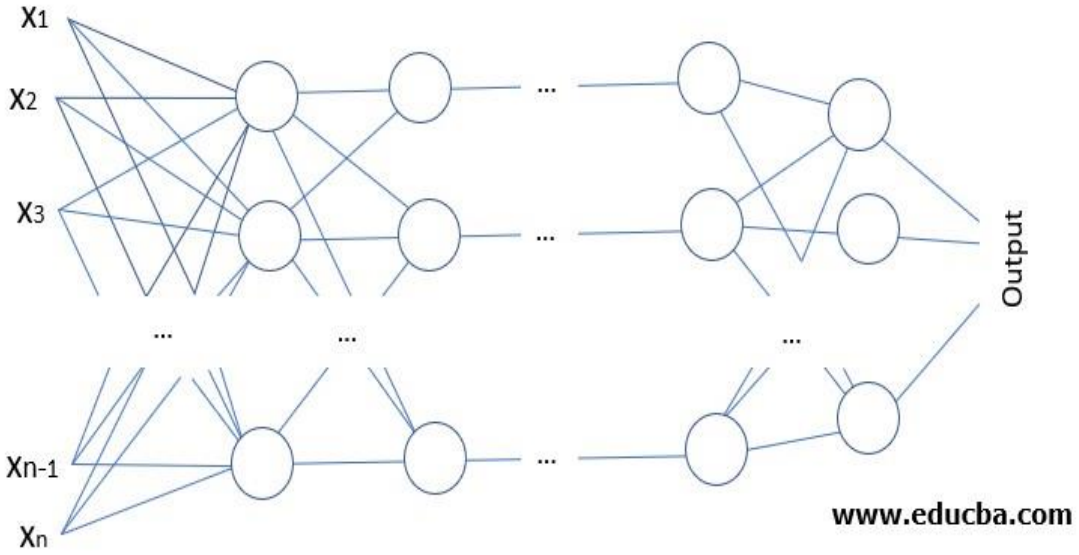



Fig:2.2 Multi-Layer Perceptron

**4. Convolution Neural Network:**

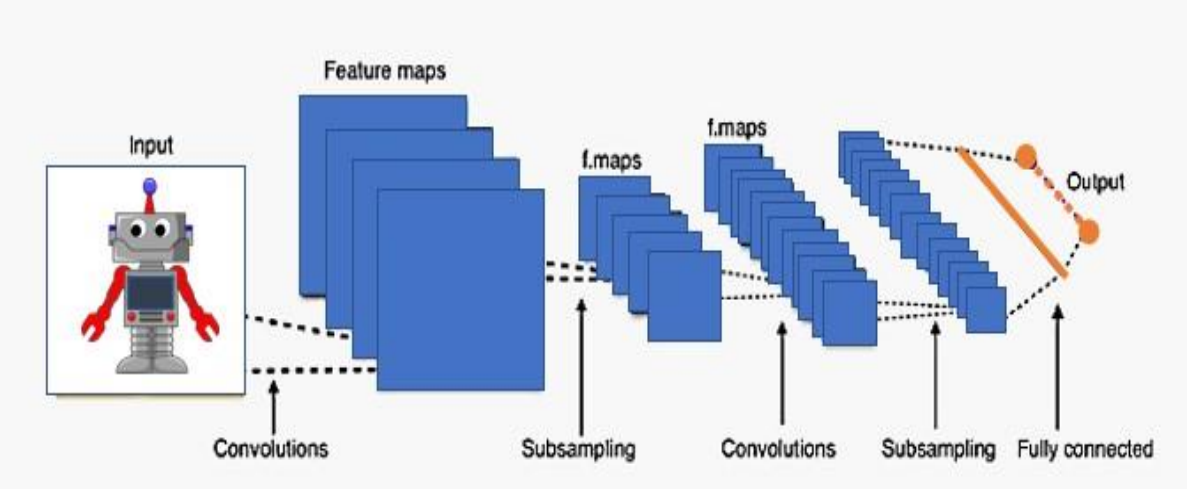



Fig:2.3 Convolution Neural Network

- CNN is one of the variations of the multilayer perceptron.
- CNN can contain more than 1 convolution layer and since it contains a convolution layer the network is very deep with fewer parameters.
- CNN is very effective for image recognition and identifying different image patterns.

**5.Recurrent Neural Network:**

- RNN is a type of neural network where the output of a particular neuron is fed back as an input to the same node.
- These method helps the network to predict the output.
- This kind of network is useful in maintaining a small state of memory which is very useful for developing the chatbot
- This kind of network is used in chatbot development and text to speech technologies.

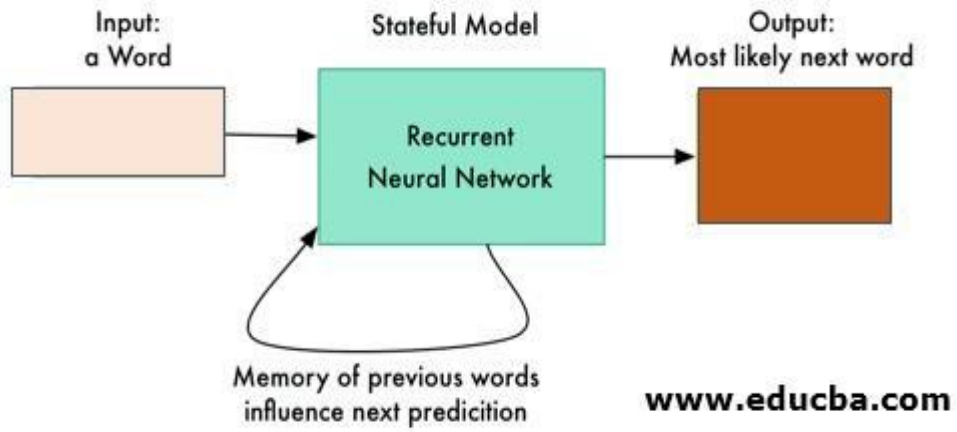



Fig:2.4 Recurrent Neural Network

**6.Modular Neural Network:**

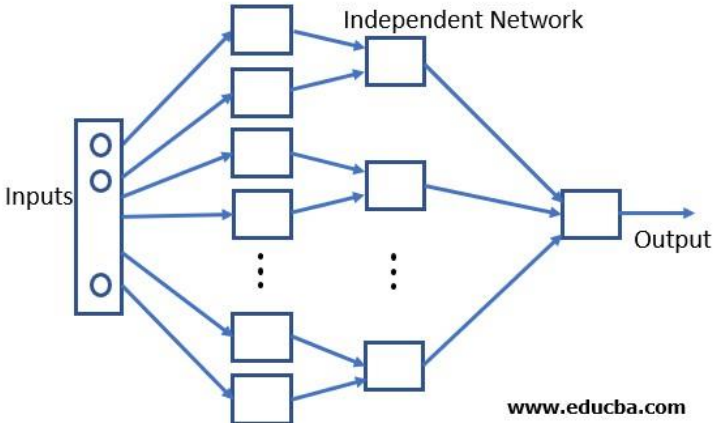



Fig:2.5 Modular Neural Network

- This kind of network is not a single network but a combination of multiple small neural networks.
- All the sub-networks make a big neural network and all of them work independently to achieve a common target.
- These networks are very helpful in breaking the small-large problem into small pieces and then solving it.

**7.Sequence to sequence models:**

- This type of network is generally a combination of two RNN networks.
- The network works on the encoding and decoding that is it consists of the encoder which is used to process the input and there is a decoder which processes the output
- Generally, this kind of network is used for text processing where the length of the inputted text is not as same as outputted text.

## 2.3 Applications of Deep Learning:

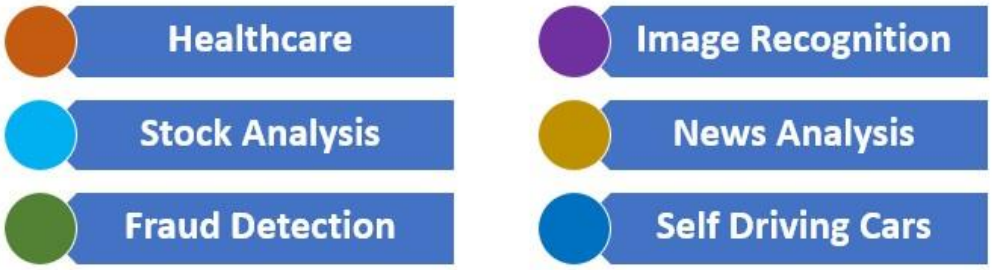



Fig:2.6 Applications of Deep Learning

1. Self Driving Cars
2. News Aggregation and Fraud News Detection
3. Natural Language Processing
4. Virtual Assistants
5. Entertainment
6. Visual Recognition
7. Fraud Detection
8. Healthcare
9. Personalisations
10. Detecting Developmental Delay in Children
11. Colourisation of Black and White images
12. Adding sounds to silent movies
13. Automatic Machine Translation
14. Automatic Handwriting Generation
15. Automatic Game Playing
16. Language Translations
17. Pixel Restoration
18. Photo Descriptions
19. Demographic and Election Predictions
20. Deep Dreaming

## 2.4 Characteristics of Deep Learning

### 1. Supervised, Semi-Supervised or Unsupervised

When the category labels are present while you train the data then it is Supervised learning. Algorithms like Linear regression. Logistic regression, decision trees use Supervised Learning. When category labels are not known while you train data then it is unsupervised learning.

Algorithms like Cluster Analysis, K means clustering, Anomaly detection uses Unsupervised Learning. The data set consists of both labeled and unlabelled data then we call it is Semi Supervised learning. Graph-based models, Generative models, cluster assumption, continuity assumption use Semi-Supervised learning.

**2.Huge Amount of Resources**

It needs advanced Graphical Processing Units for processing heavy workloads. A huge amount of data needs to be processed like Big data in the form of structured or unstructured data. Sometimes more time also required to process the data, it depends on the amount of data fed in.

**3.Large Amount of Layers in Model**

A huge amount of layers like input, activation, the output will be required, sometimes the output of one layer can be input to another layer by making few small findings and then these findings are summed up finally in the softmax layer to find out a broader classification for final output.

**4.Optimizing Hyper-parameters**

Hyper parameters like no of epochs, Batch size, No of layers, Learning rate, needs to be tuned well for successful Model accuracy because it creates a link between layer predictions to final output prediction. Over-fitting and under-fitting can be well handled with hyper-parameters.

**5.Cost Function**

It says how well the model performance in prediction and accuracy. For each iteration in Deep Learning Model, the goal is to minimize the cost when compared to previous iterations. Mean absolute error, Mean Squared Error, Hinge loss, Cross entropy are different types according to different algorithms used.

**2.5 Advantages of Deep Learning**

- Solve Complex problems like Audio processing in Amazon echo, Image recognition, etc, reduce the need for feature extraction, automated tasks wherein predictions can be done in less time using Keras and Tensorflow.
- Parallel computing can be done thus reducing overheads.
- Models can be trained on a huge amount of data and the model gets better with more data.
- High-Quality Predictions when compared with humans by training tirelessly.
- Works well-unstructured data like video clips, documents, sensor data, webcam data, etc.

## 2.6 Deep Learning Algorithms

To create a deep learning model, one must write several algorithms, blend them together and create a net of neurons. Deep learning has a high computational cost. To aid deep learning models, there are deep learning platforms like Tensor flow, Py-Torch, Chainer, Keras, etc. In deep learning, we have tried to replicate the human neural network with an artificial neural network; the human neuron is called perceptron in the deep learning model. We connect these perceptron units together to create a neural network; it has 3 sections:

1. Input layer
2. Hidden layers
3. Output layer

A perceptron has input nodes (dendrites in the human brain), an actuation function to make a small decision and output nodes (axon in the human brain). We will see how one perceptron works; connecting them together will create a deep learning model. Input information (number of input variables/features) are assigned some weight and fed to the actuation function. The actuation function makes a decision and sends output. This perceptron's output will be input to other neurons. Once the batch is processed, with backpropagation error is calculated at each neuron, with the help of a cost function cross-entropy. In this way, input weights are reassigned, and the whole process continues until cross-entropy satisfies the condition.

We have different actuation functions like Sigmoid functions, hyperbolic tangent function, Rectified Linear Unit (ReLU) to take a small decision. A deep learning model needs a vast amount of data to build a good model. Generally, a model with more than 3 hidden layers is treated as a deep neural network. Basically, Deep learning is a set of neurons with a number of

## 2.7 Architectural Methods for Deep Learning Algorithms

To build this architecture following algorithms are used:

### 1. Back Propagation

In this algorithm, we calculate partial derivatives. In general, the gradient descent method for optimization, derivatives (gradients) are calculated at each iteration. In deep learning, functions are not simple; they are the composition of different functions. In this case, it is hard to calculate gradients, so we use approximate differentiation to calculate derivatives. The more the number of parameters, the more expensive approximate differentiation will be.

**2. Stochastic Gradient Descent**

In Gradient descent, the goal is to find global minima or optimum solution. But to get that, we have to consider local minima solutions (not desirable) also. If the objective function is a convex function, it is easy to find the global minima. The initial value for the function and learning rate are deciding parameters for finding global minima. This can easily be understood by considering a river from the mountain top and searching for a foothill (global minima). But in the way, there will be some ups and downs (local minima) which must be avoided. The river originating point and speed (initial value and learning rate in our case) are deciding factors to find global minima.

**3. Learning Rate**

The learning rate is like the speed of the river; it can reduce training time and increase performance. In general, to learn any technique/sport, in the beginning, the learning rate is relatively high than at the end when one is to master it. After the intermediate stage, the learning will be slow; the focus will be on fine-tuning. The same is applied in deep learning; too large changes are tackled by a higher learning rate and by slowly decreasing the learning rate later for fine-tuning.

**4. Batch Normalization**

In deep learning initial value of weight (randomly chosen) and learning rate is defined for a minibatch. In the beginning, there would be many outliers, and during backpropagation, these outliers must be compensated to compute the weights to get output. This compensation results in extra epochs. So to avoid it, we use batch normalization.

**5. Drop Out**

In deep learning, we generally encounter the problem of overfitting. Overfitting in large networks with several parameters makes it difficult to predict on test data. So, to avoid that, we use the dropout method, which drops random units during training by creating different ‘thinned networks’. When testing these thinned networks’ predictions are averaged, which helps to avoid overfitting.

**6. Bag of Words**

We use a continuous bag of words to predict the next word. For e.g., we see in email writing the autosuggestion for completing the sentence is part of NLP. This is done by considering lots of sentences and for a specific word surrounding words that are captured. These specific words and

surrounding words are fed to the neural network. After the training model, it can predict the specific word based on the surrounding words.

**7. Long Short Term Memory**

LSTM is very useful in sequence prediction problems like language translation, predicting sales and finding the stock price. LSTM has the edge over other techniques because it is able to consider previous data. LSTM makes modification by cell states mechanism. It remembers to forget things. The 3 main aspects of LSTM make it stand out from other deep learning techniques. First is when the neuron should have input, second when to remember previous data and what to forget and third is when to pass output.

## 2.8 Libraries of Deep Learning

All the libraries which are generally used for deep learning are open source and few of them are as follows:

- TensorFlow
- deeplearning4j
- Torch
- Caffe
- Microsoft CNTK
- ML.NET
- Theano
- Deep mat
- Neon

**1. TensorFlow**

- Tensor Flow is the machine learning and deep learning library developed by Google and it came into the market around the 2016 march.
- TensorFlow grew out of an in-house library of google brain known as Dist Belief.
- Currently, TensorFlow is the leading and most used library in the market.
- Different types of deep nets can be developed and also the various packages available in this library are used to attain and address most of the tasks and problems in the field of deep learning.

**2. Deeplearning4j**

- Deeplearning4j is the open-source java library which only supports java programming language and this library is written in Java.
- This was developed by Adam Gibson to provide distributed multimode capabilities for deep neural networks.
- This library is very much use full for the application which is having build on top of big data.
- This library works with Scala and also provide inbuilt GPU support.

**3. Torch**

- This open-source deep-learning library was developed by Facebook and Twitter.
- This library is written in Lua programming language.
- However PyTorchis the library which is widely used, and it's written in a python programming language

**4.Caffe**

- Caffe is an open-source deep-learning library written in C++/CUDA and developed by Yangqing Jia of Google.
- This library was first developed for computer vision tax.
- Caffe gives permission to the user to configure the hyperparameters for a deep net.
- The layer configuration is very robust and very much sophisticated.

**5.Theano**

- This is the open-source deep-learning library written in Python and CUDA.
- This library is very similar to the TensorFlow library but the implementation and usage are not that simple as that of TensorFlow.
- This library is generally used for educational and research purposes.

**4. Microsoft CNTK**

- This is a cognitive toolkit developed by Microsoft to venture in the field of Artificial intelligence.
- This library is written in python and its supports the other packages and libraries which python programming language supports, and it comes with Microsoft visual studio.

**5. ML.NET**

- ML.NET is the open-source library which is also developed by Microsoft for the dot net developers.
- This library is written in C# and F# and it uses the Microsoft dot net platform.

**6. Deepmat**

- This library is developed in MATLAB.
- With the use of this library, we can implement deep learning using MATLAB.
- with this library GSN, CNN, Restricted Boltzmann machine, Deep belief networks, multi-layer perceptron, and many more artificial neural networks.

**7. Neon**

- Neon is a deep learning framework created by the Nervana systems to deliver industry-leading cutting edge technologies.
- This framework has been depreciated as of 2018 and further research has been carried out by Intel corporation on the same.

**Layers in Convolutional Neural Networks**

Below are the Layers of convolutional neural networks:

**1. Image Input Layer:**

The input layer gives inputs( mostly images), and normalization is carried out. Input size has to be mentioned here.

**2. Convolutional Layer:**

Convolution is performed in this layer. The image is divided into perceptrons(algorithm); local fields are created, leading to the compression of perceptrons to feature maps as a matrix with size m x n.

**3. Non-Linearity Layer:**

Here feature maps are taken as input, and activation maps are given as output with the help of the activation function. The activation function is generally implemented as sigmoid or hyperbolic tangent functions.

**4. Rectification Layer:**

The crucial component of CNN, this layer does the training faster without reducing accuracy. It performs element-wise absolute value operation on activation maps.

**5. Rectified Linear Units(ReLU):**

ReLU combines non-linear and rectification layers on CNN. This does the threshold operation where negative values are converted to zero. However, ReLU doesn't change the size of the input.

**6. Pooling Layer:**

The pooling layer is also called the down sampling layer, as this is responsible for reducing the size of activation maps. A filter and stride of the same length are applied to the input volume. This layer ignores less significant data; hence image recognition is done in a smaller representation. This layer reduces overfitting. Since the amount of parameters is reduced using the pooling layer, the cost is also reduced. The input is divided into rectangular pooling regions, and either maximum or average is calculated, which returns maximum or average consequently. Max Pooling is a popular one.

**7. Dropout Layer:**

This layer randomly sets the input layer to zero with a given probability. More results in different elements are dropped after this operation. This layer also helps to reduce overfitting. It makes the network to be redundant. No learning happens in this layer. This operation is carried out only during training.

**8. Fully Connected Layer:**

Activation maps, which are the output of previous layers, is turned into a class probability distribution in this layer. FC layer multiplies the input by a weight matrix and adds the bias vector.

**9. Output Layer:**

FC layer is followed by soft max and classification layers. The soft max function is applied to the input. The classification layer computes the cross-entropy and loss function for classification problems.

**10. Regression Layer:**

Half the mean squared error is computed in this layer. This layer should follow the FC layer.

# 3. SYSTEM ANALYSIS

## 3.1 Existing System

The previous work on “Shape matching and object recognition using shape contexts,” and “ANSIG—An analytic signature for permutation-invariant two dimensional shape representation,” have used different global descriptors of the full logo image either accounting for logo contours or exploiting shape descriptors such as shape context. A two-stage algorithm proposed in “Logo detection based on spatial-spectral saliency and partial spatial context,” that accounts for local contexts of key points. They considered spatial-spectral saliency to avoid the impact of cluttered background and speed up the logo detection and localization. Appropriate metrics is accomplished among available methods, using two publicly available fundus datasets. In addition, the paper proposed the comprehensive normalization method which recorded acceptable results when applied for color normalization. The drawback of this method is that it assumes that a logo picture is fully visible in the image, is not corrupted by noise and is not subjected to transformations. According to this, they cannot be applied to real world images. The major limitation of this approach is image resolution and their solution has revealed to be very sensitive to occlusions.

**Disadvantages of Existing System:**

- Less Accuracy
- Cannot be applied to the real world images.
- It is not corrupted by the noise.

## 3.2 Proposed System

In this project we have used CNN (convolution neural network) algorithm to classify logo as fake or original. To Train this algorithm we have used below logo images. CNN is a type of deep learning model for processing data that has a grid pattern, such as images, which is inspired by the organization of animal visual cortex and designed to automatically and adaptively learn spatial hierarchies of features, from low- to high-level patterns.

**Advantages of Proposed System:**

- High Accuracy
- Enables the monitoring of logo variants on products.
- Allows brands to issue brands to issue take downs.

**Modules Information:**

To implement this project we have designed following modules

1) Upload Logo Dataset: using this module we will upload dataset to application.
2) Pre-process Dataset: using this module we will read each image and then resize all images to equal size and then normalize pixel values and then shuffle dataset. After processing we will split dataset into train and test where application using 80% dataset images for training and 20% for testing.
3) Train CNN Algorithm: using this module we will input 80% training images to CNN algorithm to train a model and this model will be applied on 20% test images to calculate prediction accuracy.
4) CNN Training Graph: using this module we will plot CNN training accuracy and loss graph.
5) Logo Classification: using this module we will upload test image and then CNN will classify those images as Fake or original.

# FUNCTIONAL REQUIREMENTS:

## SOFTWARE REQIREMENTS:

### System Attributes:

1. filename
2. classifier
3. labels, X, Y, X_train, Y_train, X_test,Y_test, classifier

### Data base Requirements:

No need

## USECASE:

Use cases - Use cases describe the interaction between the system and external users that leads to achieving particular goals.

To implement this project we have designed following modules

- Upload Logo Dataset
- Pre-process Dataset
- Train CNN Algorithm
- CNN Training Graph
- Logo Classification

**User Stories:** In this project we have used CNN (convolution neural network) algorithm to classify logo as fake or original. To Train this algorithm we have used below logo images.

**Prototype:**

python 3.7.0 or 3.7.4

opencv-python==4.5.1.48

keras==2.3.1

tensorflow==1.14.0

protobuf==3.16.0

h5py==2.10.0

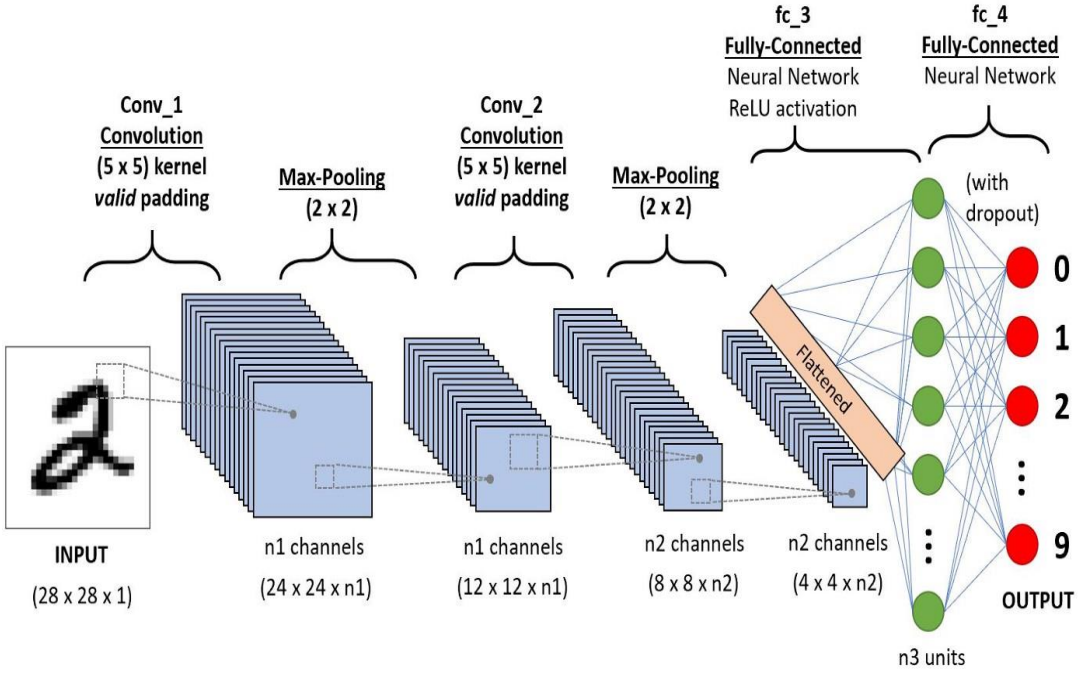
sklearn-extensions==0.0.2

scikit-learn==0.22.2.post1

Numpy

Pandas

**Models and Diagrams:**



**NONFUNCTIONAL REQUIREMENT:**

**Usability:** Usability is a quality attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process.(how it was handle entire project easy)

**Security:** The quality or state of being secure: such as. a : freedom from danger : safety. b : freedom from fear or anxiety. c : freedom from the prospect of being laid off job security.

**Readability:** Readability is the ease with which a reader can understand a written text.

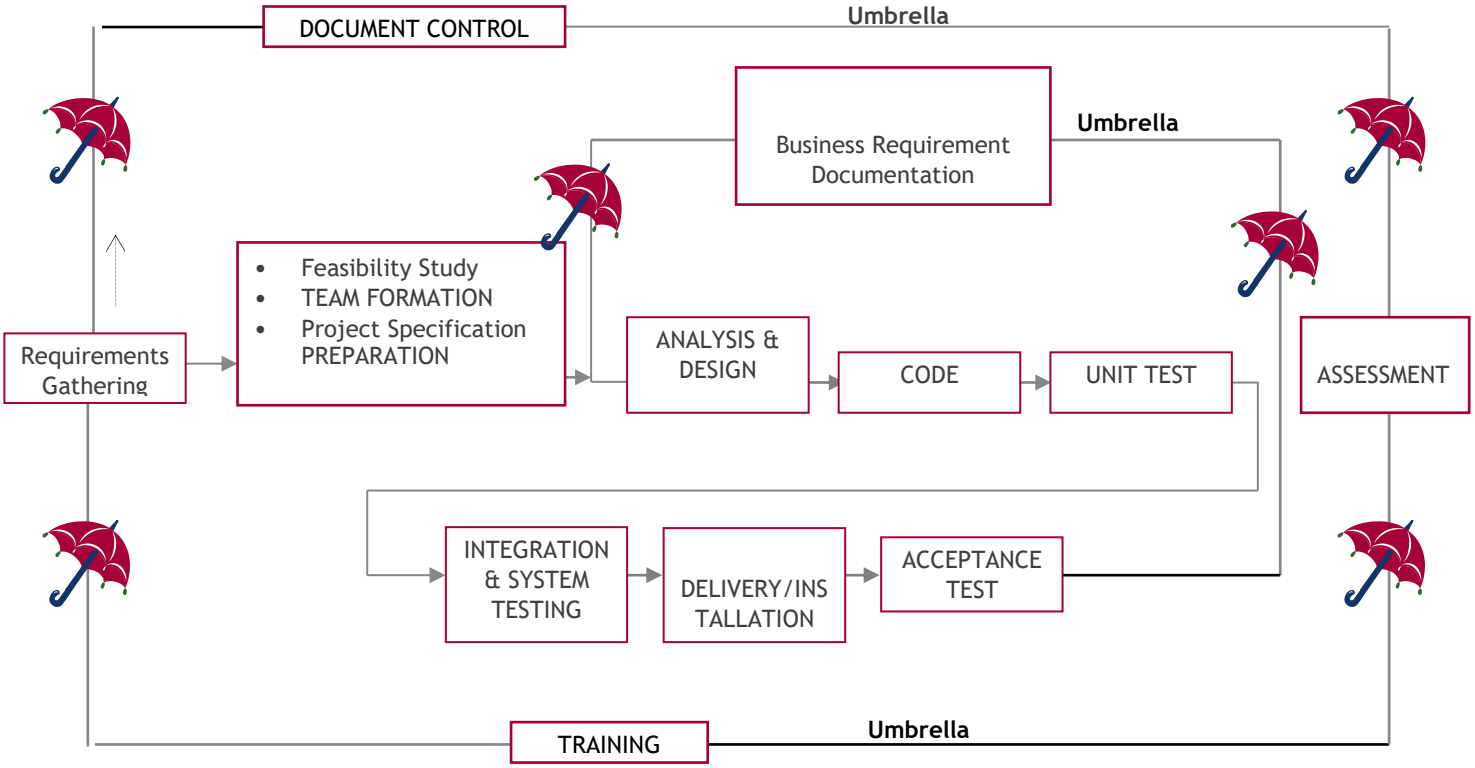
**Performance**: The execution of an action. : something accomplished : deed, feat. : the fulfillment of a claim, promise, or request : implementation. 3. : the action of representing a character in a play.

**Availability**: The quality or state of being available trying to improve the availability of affordable housing. 2 : an available person or thing.

**Scalability**: Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands.

## 3.3. PROCESS MODEL USED WITH JUSTIFICATION

**SDLC (Umbrella Model):**




SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

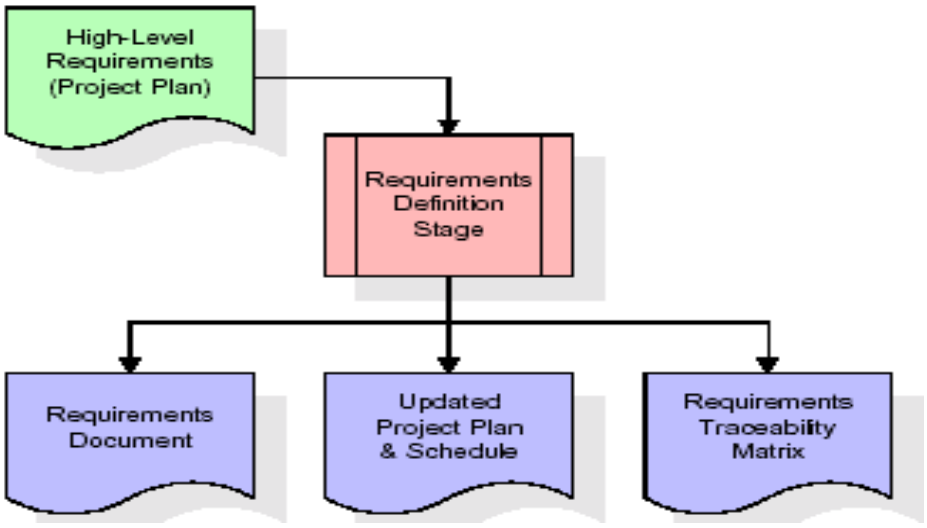
**Stages in SDLC:**

- Requirement Gathering
- Analysis
- Designing
- Coding
- Testing
- Maintenance

**Requirements Gathering stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define

operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.

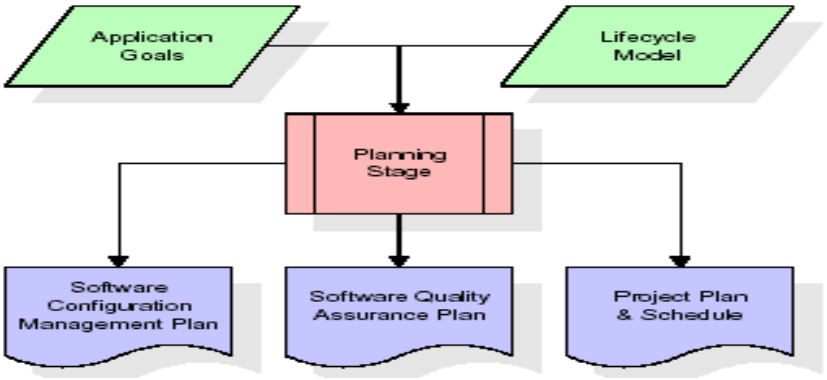



The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

- Feasibility study is all about identification of problems in a project.
- No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
- Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

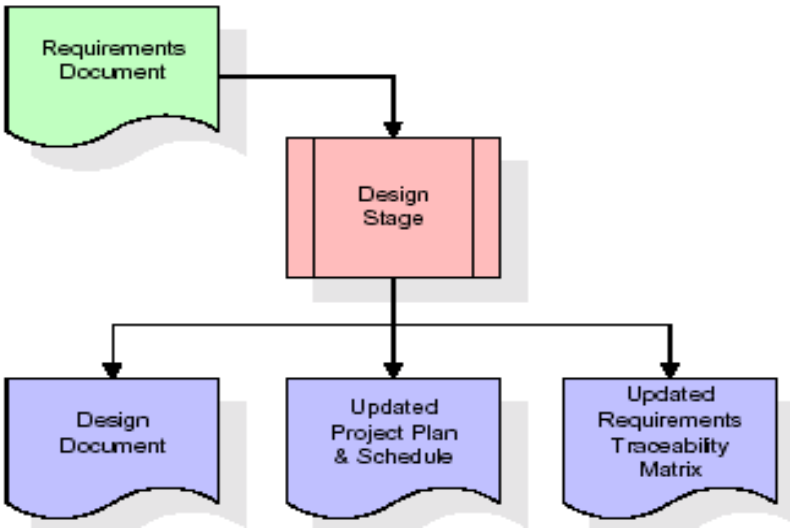
The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.




The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included.

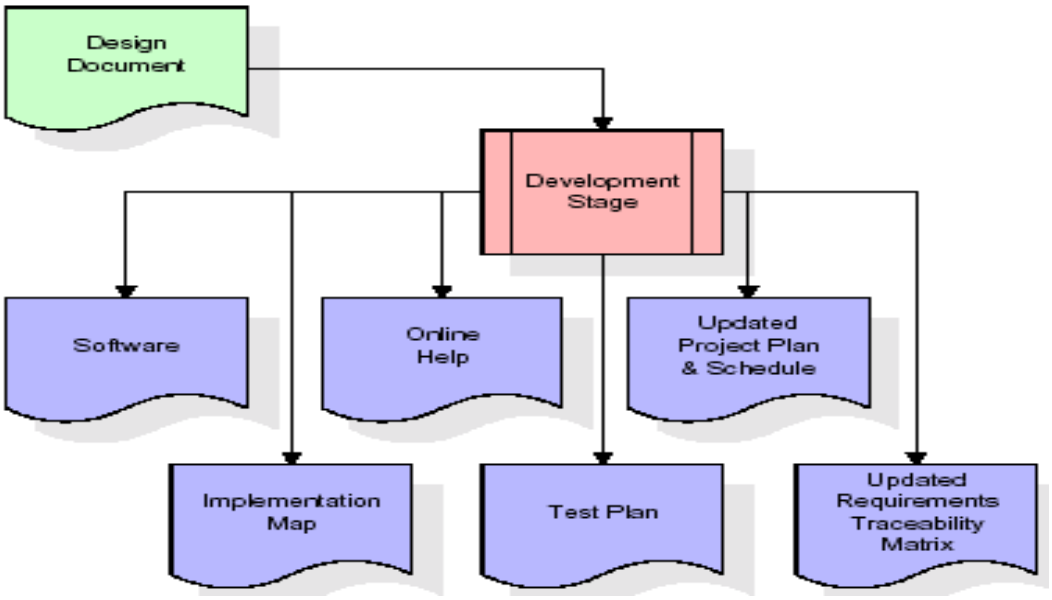
**Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.



**Development (Coding) Stage:**

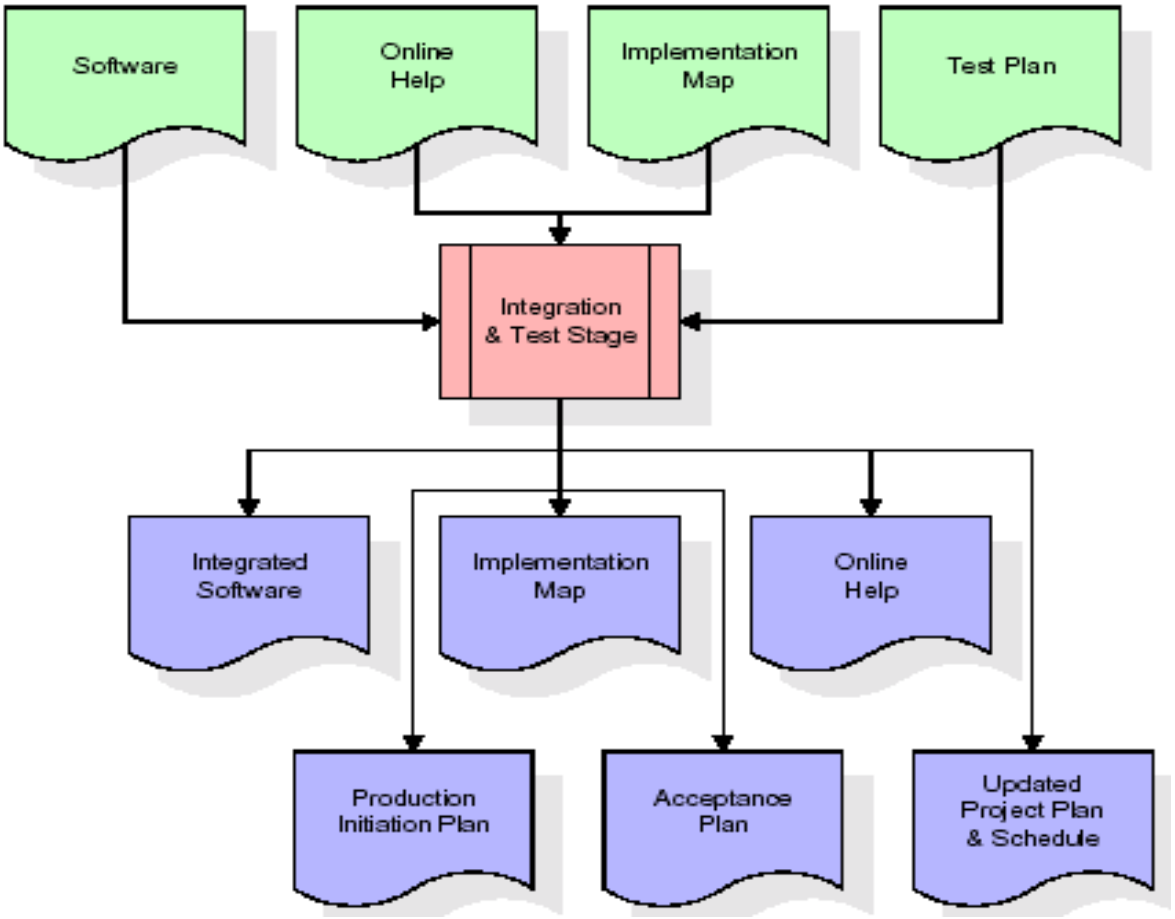
The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed.




The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

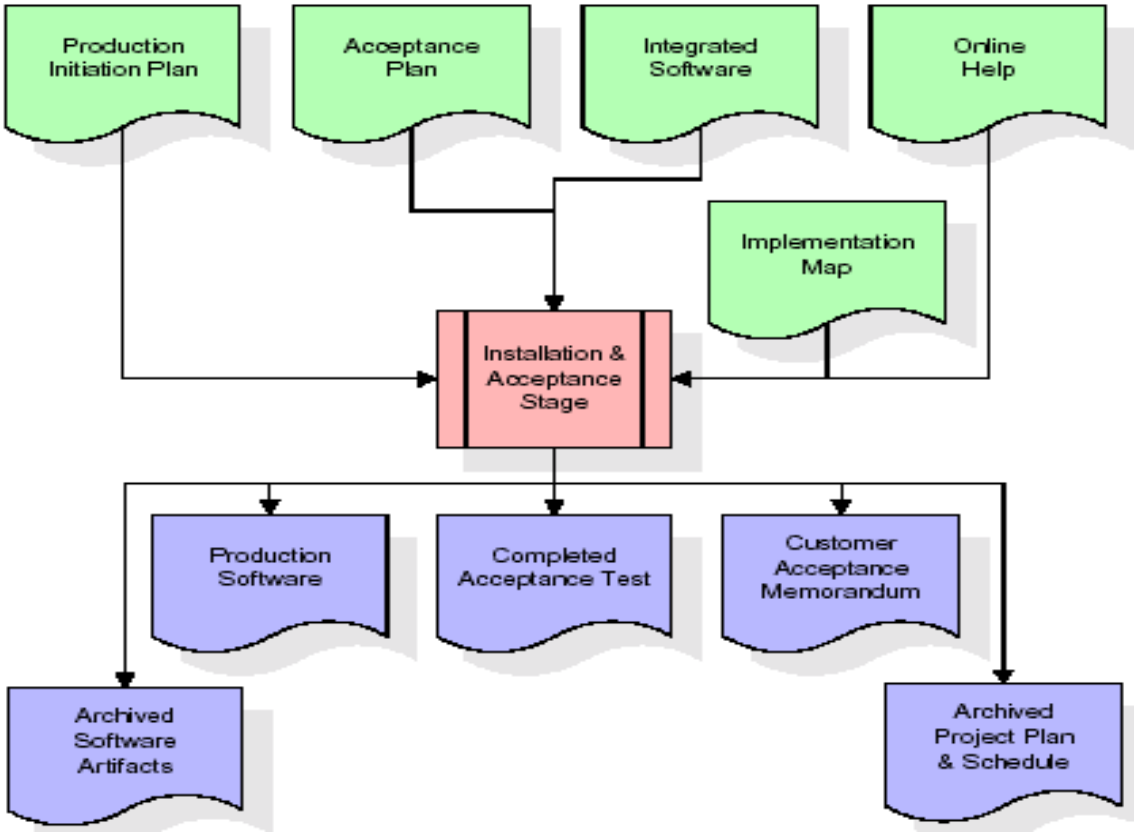



The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

- **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.




**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

## 3.4. Software Requirement Specification

### 3.4.1. Overall Description

A Software Requirements Specification (SRS) – a requirements specification for a software system is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Nonfunctional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose

solutions. Within the systems development lifecycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

- Business requirements describe in business terms what must be delivered or accomplished to provide value.

- Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)

- Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation.

**3.5 SYSTEM REQUIREMENT:**

**HARDWARE REQUIREMENTS:**

- Processor - Intel i3(min)
- Speed - 1.1 GHz
- RAM - 4GB(min)
- Hard Disk - 500 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse

**SOFTWARE REQUIREMENTS:**

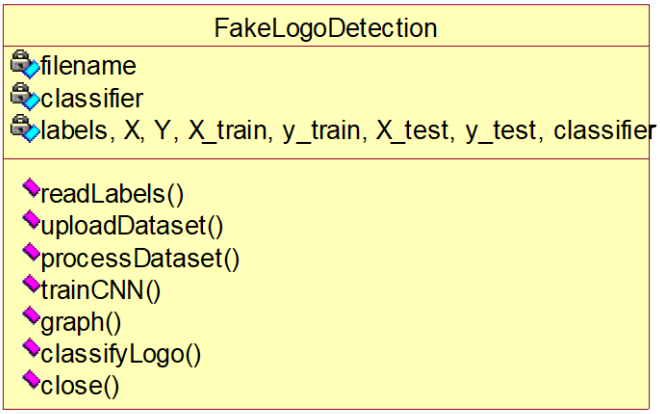
- Operating System - Windows10(min)
- Programming Language - Python

# 4.SYSTEM DESIGN

## CLASS DIAGRAM:

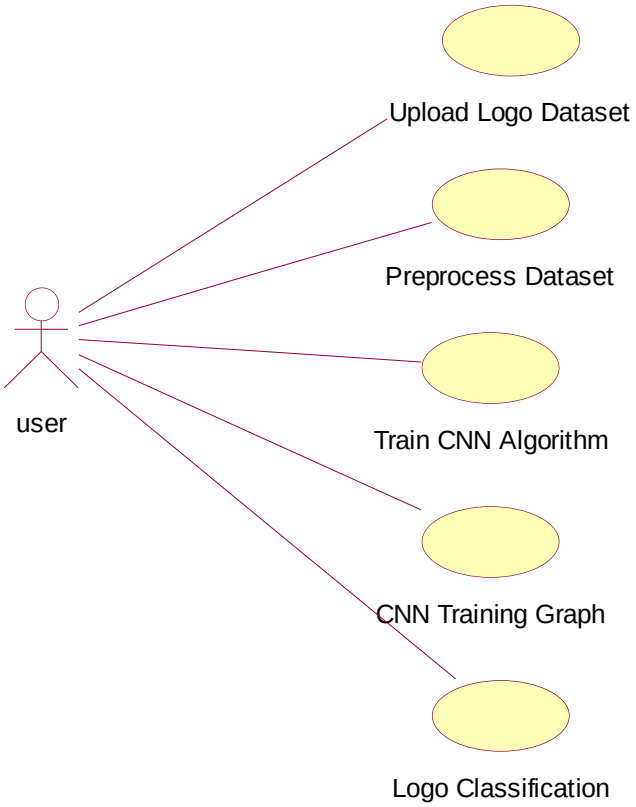
The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

- The upper part holds the name of the class
- The middle part contains the attributes of the class
- The bottom part gives the methods or operations the class can take or undertake



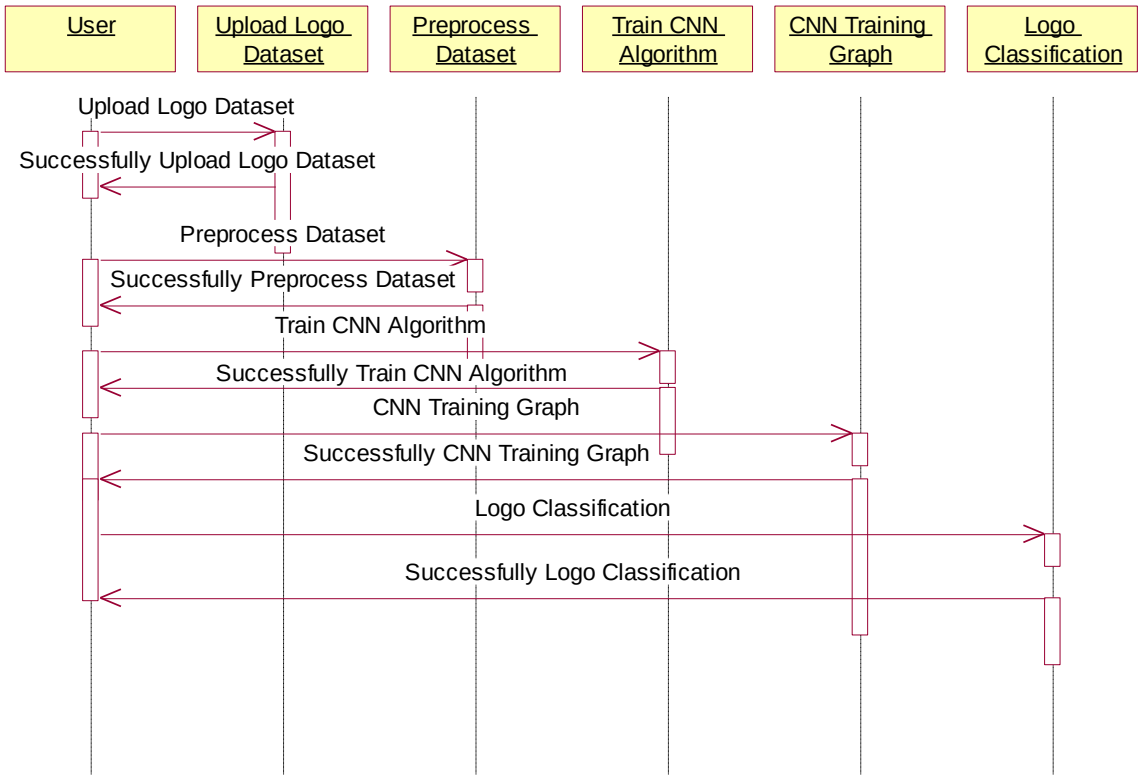

USECASE DIAGRAM**:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as we




**SEQUENCE DIAGRAM**

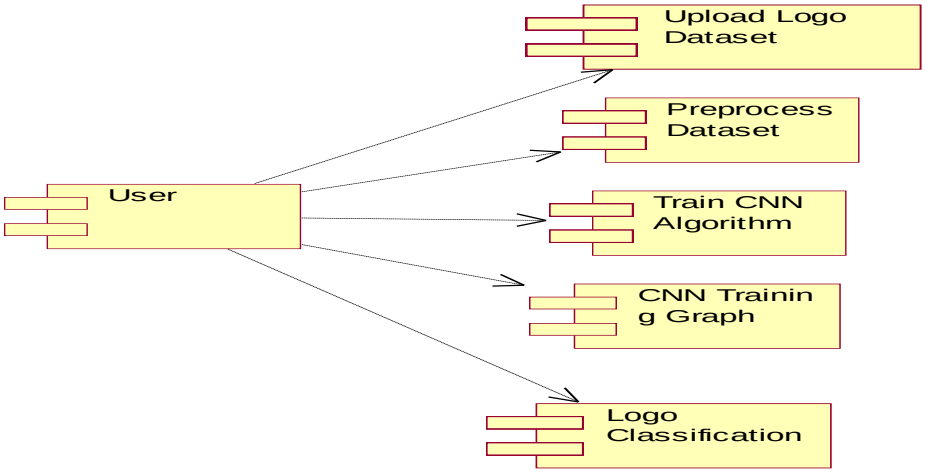
A **sequence diagram** is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams**, **event scenarios**, and timing diagrams.




**COMPONENT DIAGRAM:**

In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

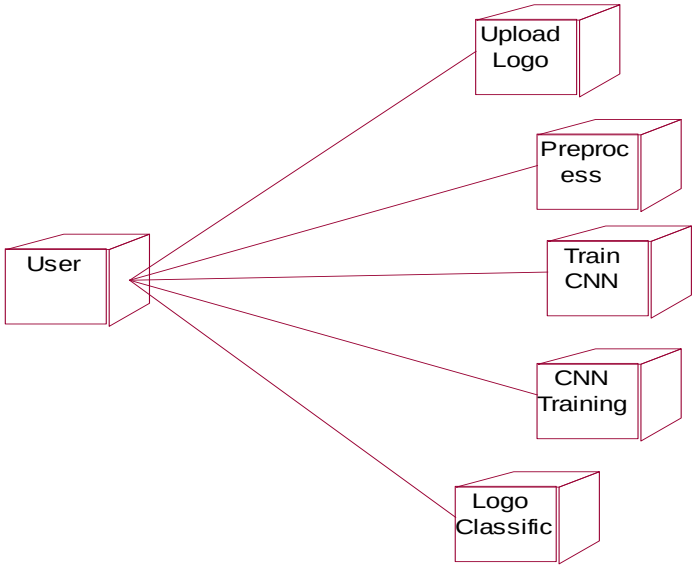
Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.




**DEPLOYMENT DIAGRAM:**

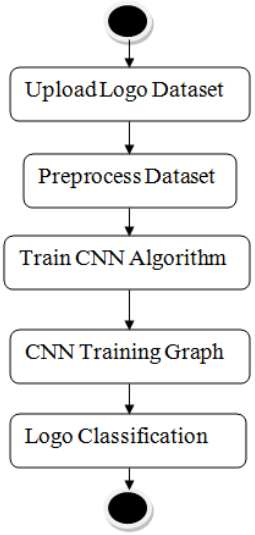
A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in deployment diagram may conceptually represent multiple physical nodes, such as a cluster of deployment diagram may conceptually represent multiple physical nodes, such as a cluster of data.



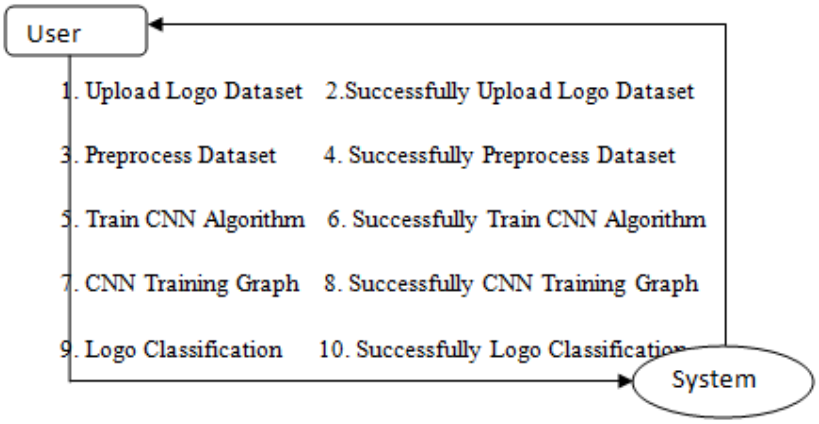
**ACTIVITY DIAGRAM:**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent




**Data flow :**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function.



# 5.IMPLEMENTATION

## 5.1 Python

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

## 5.2 Sample Code:

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
from tkinter.filedialog import askopenfilename
import cv2
import random
import numpy as np
from keras.utils.np_utils import to_categorical
from keras.layers import  MaxPooling2D
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Convolution2D
from keras.models import Sequential
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from keras.callbacks import ModelCheckpoint
import pickle
import os
from keras.models import load_model
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
```

```
from sklearn.metrics import confusion_matrix
import seaborn as sns

main = tkinter.Tk()
main.title("Fake Log Detection")
main.geometry("1300x1200")

global filename
global classifier
global labels, X, Y, X_train, y_train, X_test, y_test, classifier

def readLabels(filename):
    global labels
    labels = []
    for root, dirs, directory in os.walk(filename):
        for j in range(len(directory)):
            name = os.path.basename(root)
            if name not in labels:
labels.append(name)

def uploadDataset():
    global filename
    global labels
    labels = []
    filename = filedialog.askdirectory(initialdir=".")
pathlabel.config(text=filename)
text.delete('1.0', END)
text.insert(END,filename+" loaded\n\n");
readLabels(filename)
text.insert(END,"Logo found in dataset are\n\n")
    for i in range(len(labels)):
text.insert(END,labels[i]+"\n")

def processDataset():
text.delete('1.0', END)
```

```
    global filename, X, Y, X_train, y_train, X_test, y_test
    if os.path.exists("model/X.txt.npy"):
        X = np.load('model/X.txt.npy')
        Y = np.load('model/Y.txt.npy')
    else:
        for root, dirs, directory in os.walk(filename):
            for j in range(len(directory)):
                name = os.path.basename(root)
                if 'Thumbs.db' not in directory[j]:
img = cv2.imread(root+"/"+directory[j])
img = cv2.resize(img, (64,64))
                    im2arr = np.array(img)
                    im2arr = im2arr.reshape(64,64,3)
X.append(im2arr)
                    label = getID(name)
Y.append(label)
print(name+" "+str(label))
        X = np.asarray(X)
        Y = np.asarray(Y)
np.save('model/X.txt',X)
np.save('model/Y.txt',Y)
    X = X.astype('float32')
    X = X/255
text.insert(END,"Dataset Preprocessing Completed\n")
text.insert(END,"Total images found in dataset : "+str(X.shape[0])+"\n\n")
    indices = np.arange(X.shape[0])
np.random.shuffle(indices)
    X = X[indices]
    Y = Y[indices]
    Y = to_categorical(Y)
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2) #split dataset into train and test
text.insert(END,"80% images are used to train CNN : "+str(X_train.shape[0])+"\n")
text.insert(END,"20% images are used to train CNN : "+str(X_test.shape[0])+"\n")
```

```
def trainCNN():
text.delete('1.0', END)
    global filename, X, Y, X_train, y_train, X_test, y_test, classifier, labels
    classifier = Sequential()
classifier.add(Convolution2D(32, (3, 3), input_shape = (X_train.shape[1], X_train.shape[2], X_train.shape[3]), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (1, 1)))
classifier.add(Convolution2D(32, (3, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (1, 1)))
classifier.add(Flatten())
classifier.add(Dense(units = 256, activation = 'relu'))
classifier.add(Dense(units = y_train.shape[1], activation = 'softmax'))
classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
    if os.path.exists("model/model_weights.hdf5") == False:
model_check_point = ModelCheckpoint(filepath='model/model_weights.hdf5', verbose = 1, save_best_only = True)
        hist = classifier.fit(X, Y, batch_size = 32, epochs = 20, validation_data=(X_test, y_test), callbacks=[model_check_point], verbose=1)
    else:
        classifier = load_model("model/model_weights.hdf5")
    predict = classifier.predict(X)
    predict = np.argmax(predict, axis=1)
testY = np.argmax(Y, axis=1)
    p = precision_score(testY, predict,average='macro') * 100
    r = recall_score(testY, predict,average='macro') * 100
    f = f1_score(testY, predict,average='macro') * 100
    a = accuracy_score(testY,predict)*100
text.insert(END,"CNN Accuracy  : "+str(a)+"\n")
text.insert(END,"CNN Precision : "+str(p)+"\n")
text.insert(END,"CNN Recall    : "+str(r)+"\n")
text.insert(END,"CNN FSCORE    : "+str(f)+"\n\n")
conf_matrix = confusion_matrix(testY, predict)
plt.figure(figsize =(6, 6))
    ax = sns.heatmap(conf_matrix, xticklabels = labels, yticklabels = labels, annot = True, cmap="viridis" ,fmt ="g");
```

```
ax.set_ylim([0,len(labels)])
plt.title("CNN Confusion matrix")
plt.ylabel('True class')
plt.xlabel('Predicted class')
plt.show()

def graph():
    f = open('model/history.pckl', 'rb')
    graph = pickle.load(f)
f.close()
    accuracy = graph['val_accuracy']
    error = graph['val_loss']
plt.figure(figsize=(10,6))
plt.grid(True)
plt.xlabel('EPOCH')
plt.ylabel('Accuracy/Loss')
plt.plot(accuracy, 'ro-', color = 'green')
plt.plot(error, 'ro-', color = 'red')
plt.legend(['CNN Accuracy', 'CNN Loss'], loc='upper left')
plt.title('CNN Training Accuracy & Loss Graph')
plt.show()

def classifyLogo():
    global classifier, labels
    filename = filedialog.askopenfilename(initialdir="testImages")
    image = cv2.imread(filename)
img = cv2.resize(image, (64, 64))
    im2arr = np.array(img)
    im2arr = im2arr.reshape(1,64,64,3)
img = np.asarray(im2arr)
img = img.astype('float32')
img = img/255
    preds = classifier.predict(img)
    predict = np.argmax(preds)
```

```
img = cv2.imread(filename)
img = cv2.resize(img, (700,400))
    cv2.putText(img, 'Logo Predicted as : '+labels[predict], (10, 25), cv2.FONT_HERSHEY_SIMPLEX,0.7, (0, 0, 255), 2)
    cv2.imshow('Logo Predicted as : '+labels[predict], img)
    cv2.waitKey(0)

def close():
main.destroy()
    font = ('times', 16, 'bold')
title = Label(main, text='Fake Log Detection',anchor=W, justify=CENTER)
title.config(bg='LightGoldenrod1', fg='medium orchid')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)
font1 = ('times', 13, 'bold')
upload = Button(main, text="Upload Logo Dataset", command=uploadDataset)
upload.place(x=50,y=100)
upload.config(font=font1)
pathlabel = Label(main)
pathlabel.config(bg='yellow4', fg='white')
pathlabel.config(font=font1)
pathlabel.place(x=50,y=150)
processButton = Button(main, text="Preprocess Dataset", command=processDataset)
processButton.place(x=50,y=200)
processButton.config(font=font1)

trainButton = Button(main, text="Train CNN Algorithm", command=trainCNN)
trainButton.place(x=50,y=250)
trainButton.config(font=font1)

graphButton = Button(main, text="CNN Training Graph", command=graph)
graphButton.place(x=50,y=300)
graphButton.config(font=font1)
```

```
classifyButton = Button(main, text="Logo Classification", command=classifyLogo)
classifyButton.place(x=50,y=350)
classifyButton.config(font=font1)

exitButton = Button(main, text="Exit", command=close)
exitButton.place(x=50,y=400)
exitButton.config(font=font1)

font1 = ('times', 12, 'bold')
text=Text(main,height=25,width=78)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=370,y=100)
text.config(font=font1)
main.config(bg='burlywood2')
main.mainloop()
```

# 6. TESTING

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

**Implementation**

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user.

The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

**Testing**

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

## System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it

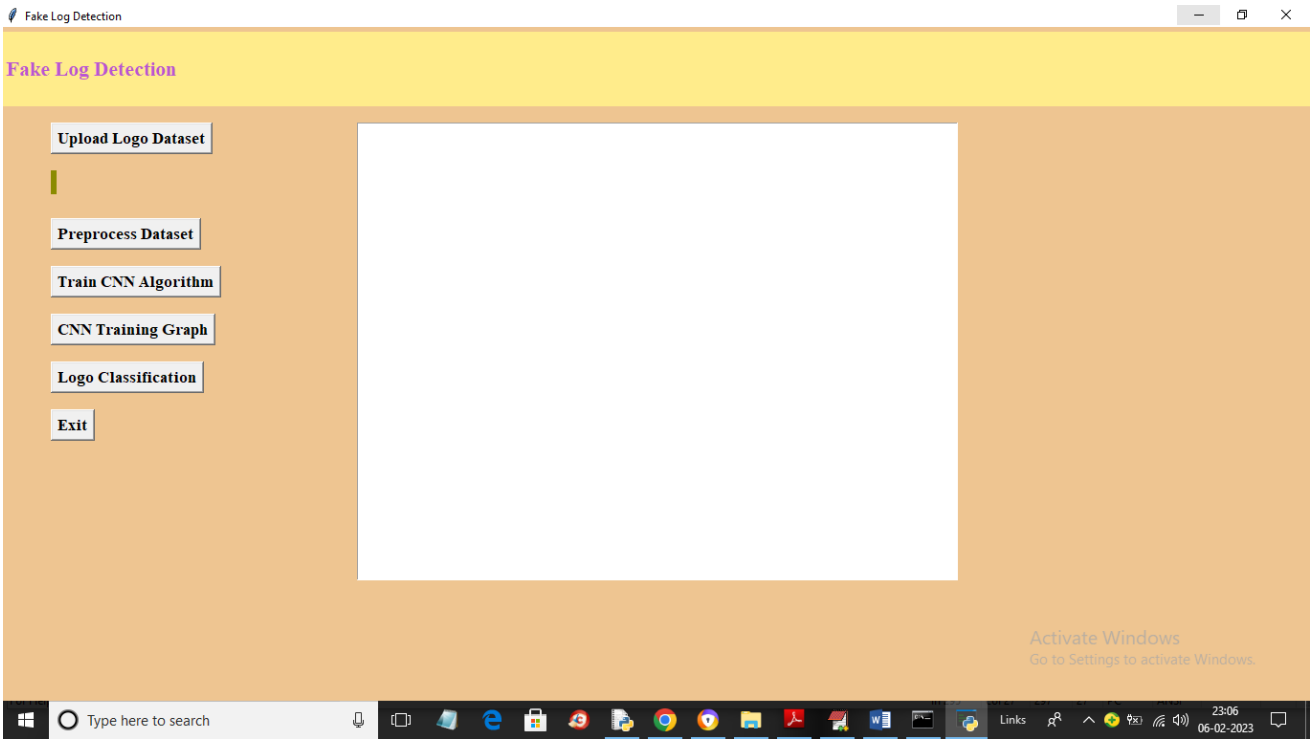
to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to use the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | **Test Case Status** | **Test Priority** |
|---|---|---|---|---|---|---|---|
| | | | **Step** | **Expected** | **Actual** | | |
| 01 | Upload Logo Dataset | Verify Logo Dataset Uploaded or not | If Logo Dataset may not upload | we cannot do any further operations | we can do further operations | High | High |
| 02 | Preprocess Dataset | Verify Preprocess Dataset or not | If Dataset may not Preprocessed | we cannot do any further operations | we can do further operations | High | High |
| 03 | Train CNN Algorithm | Verify Train CNN Algorithm or not | If Train CNN Algorithm not be | we cannot do any further operations | we can do further operations | High | High |
| 04 | CNN Training Graph | Verify CNN Training Graph or not | If CNN Training Graph not Run | We cannot run Operation | We can Run the Operation | High | High |
| 05 | Logo Classification | Verify Logo Classification or not | If Logo Classification not be | we cannot do any further operations | we can do further operations | High | High |

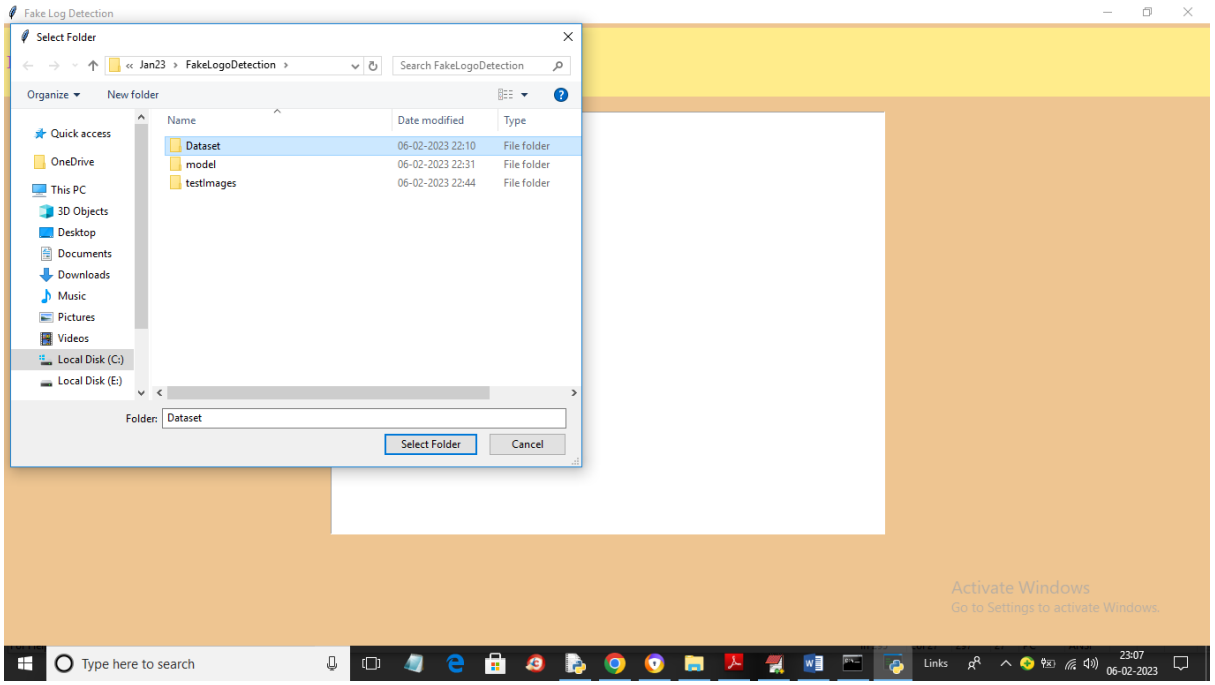
# 7. SCREENSHOTS

## Fake Logo Detection

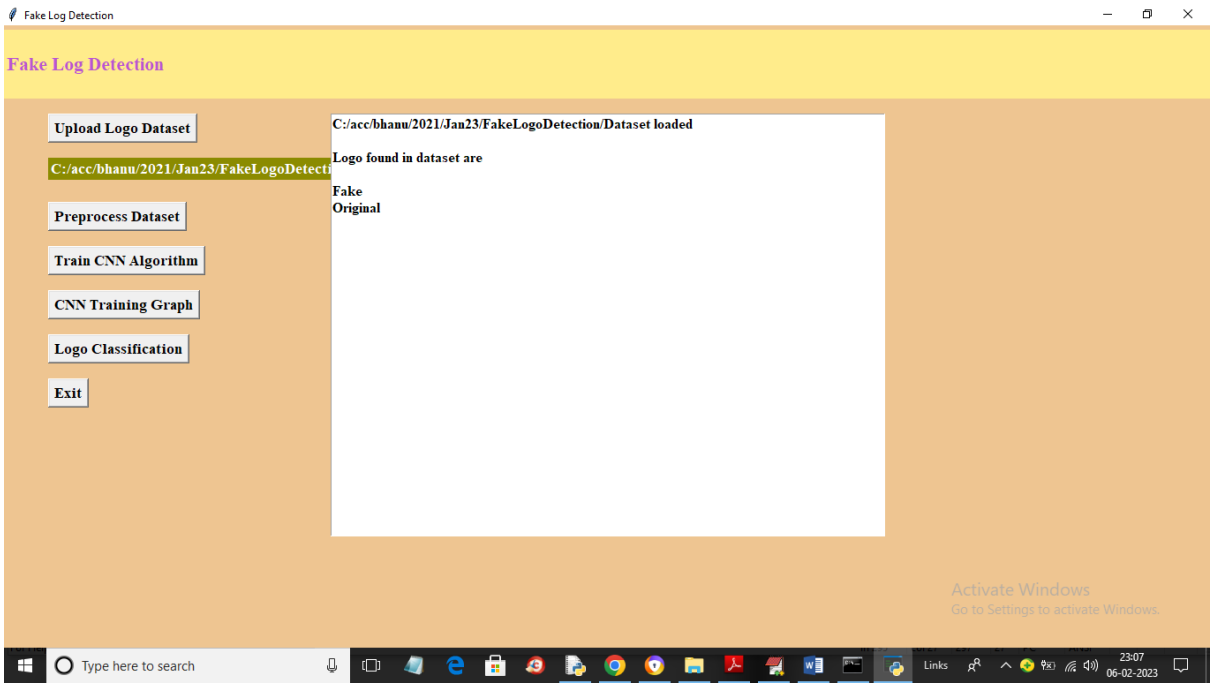
In this project we have used CNN (convolution neural network) algorithm to classify logo as fake or original. To Train this algorithm we have used below logo images.



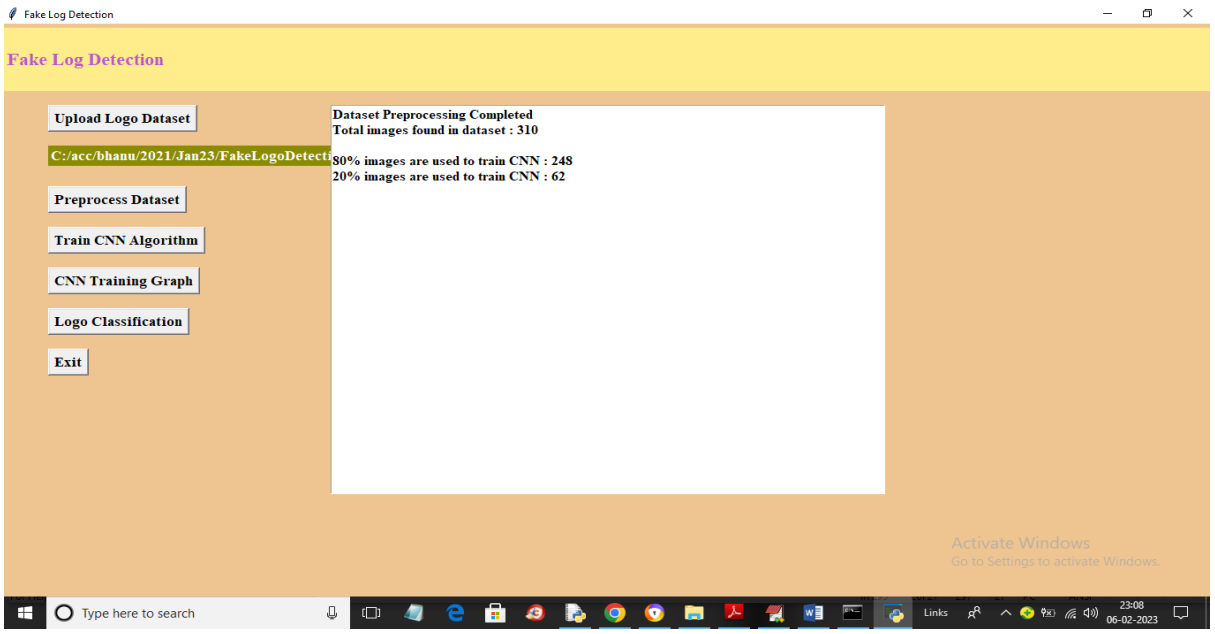

In above screen click on 'Upload Logo Dataset' button to upload dataset and get below output.



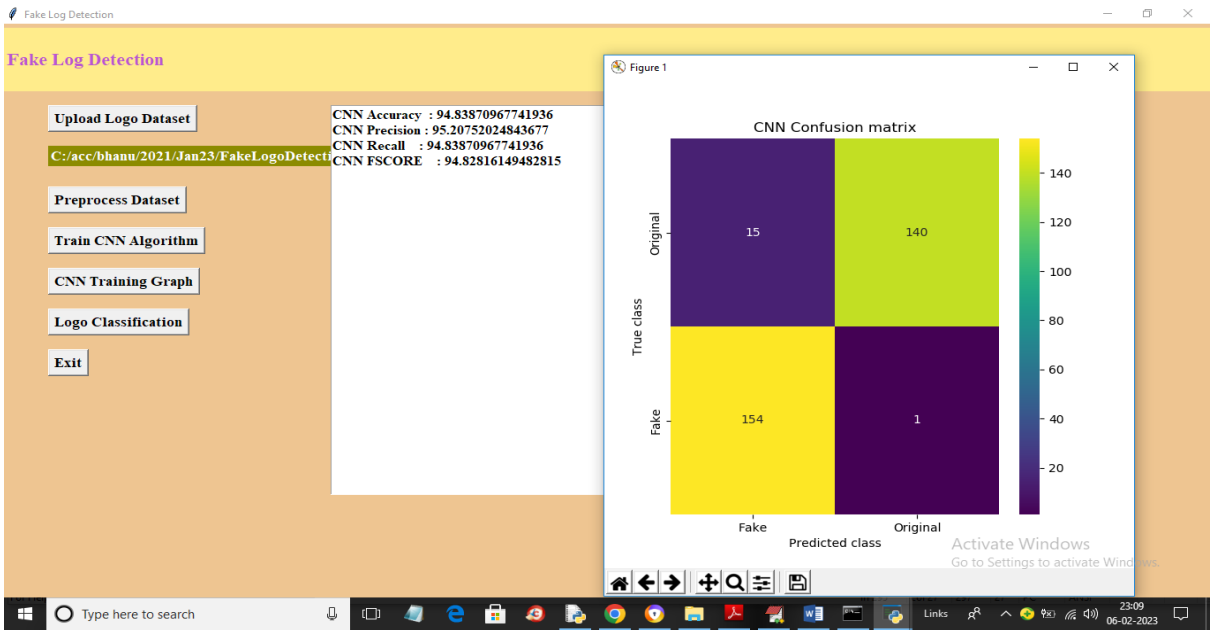

In above screen selecting and uploading entire “Dataset” folder and then click on ‘Select Folder’ button to load dataset and get below output.



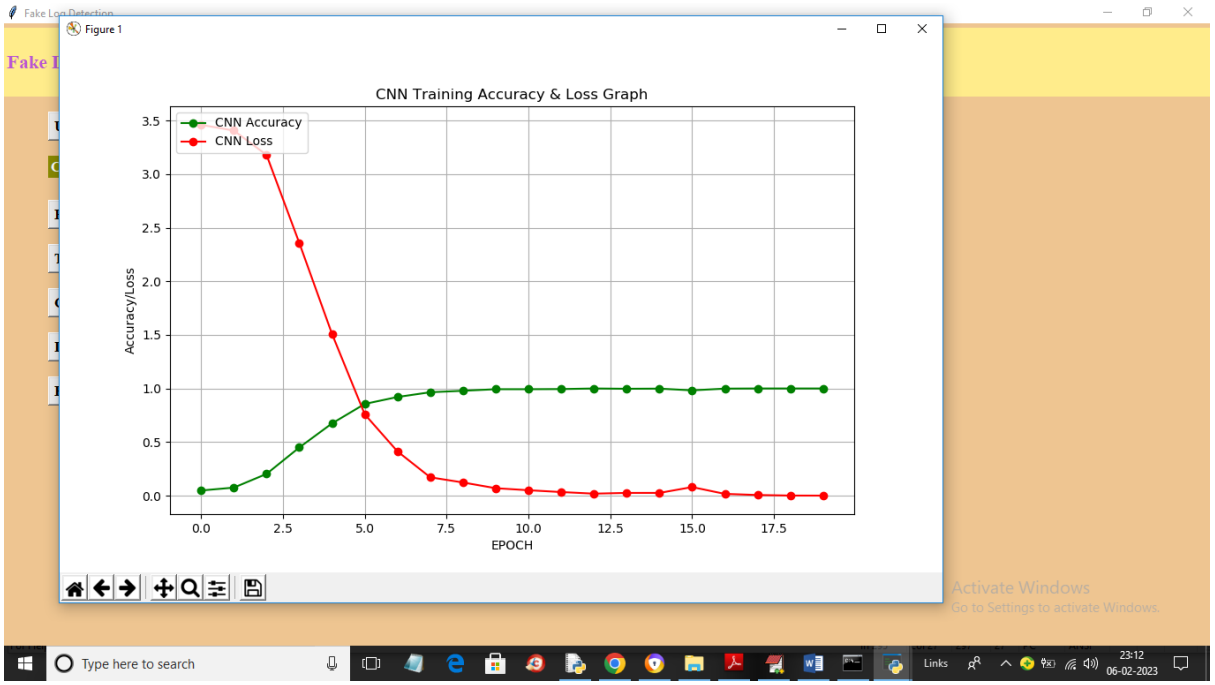

In above screen dataset images loaded as Fake and original and now click on ‘Pre-process Dataset’ button to process images and get below output.



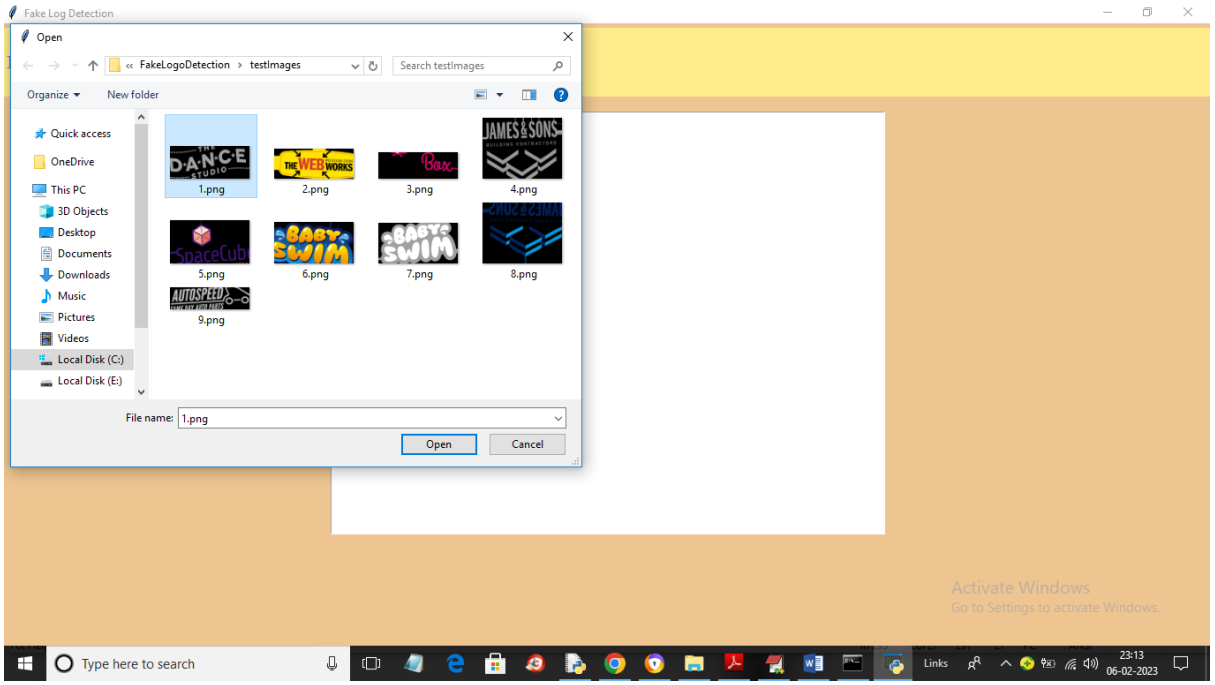

In above screen dataset processing completed and we can see dataset contains 310 images and application using 80% (248) images for training and 20% (62) images for testing and now click on 'Train CNN Algorithm' button train CNN with training images and get below output.



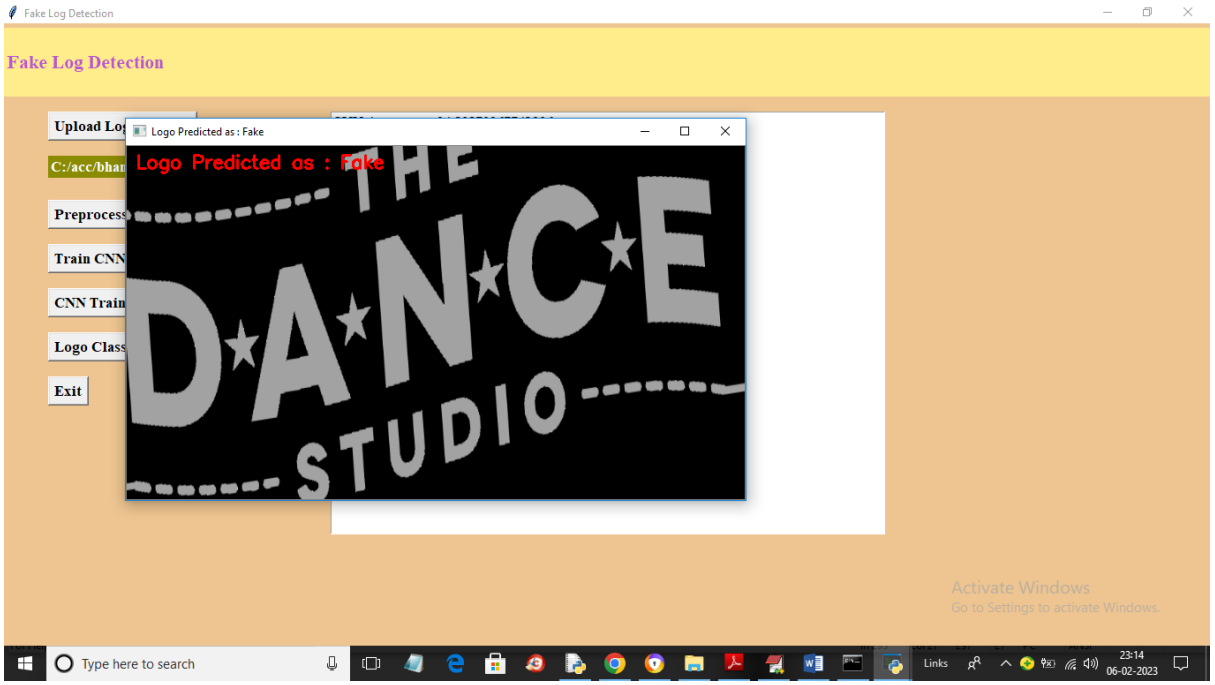

In above screen CNN training completed and we got its prediction accuracy as 94% and we can see other metric like precision, recall and FSCORE. In confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and blue colour boxes contains incorrect prediction count and different colour boxes contains correct prediction count and in above graph we can see wrong prediction counts are very few so we can say CNN predictions are more than 90% correct. Now click on 'CNN Training Graph' button to get below graph.



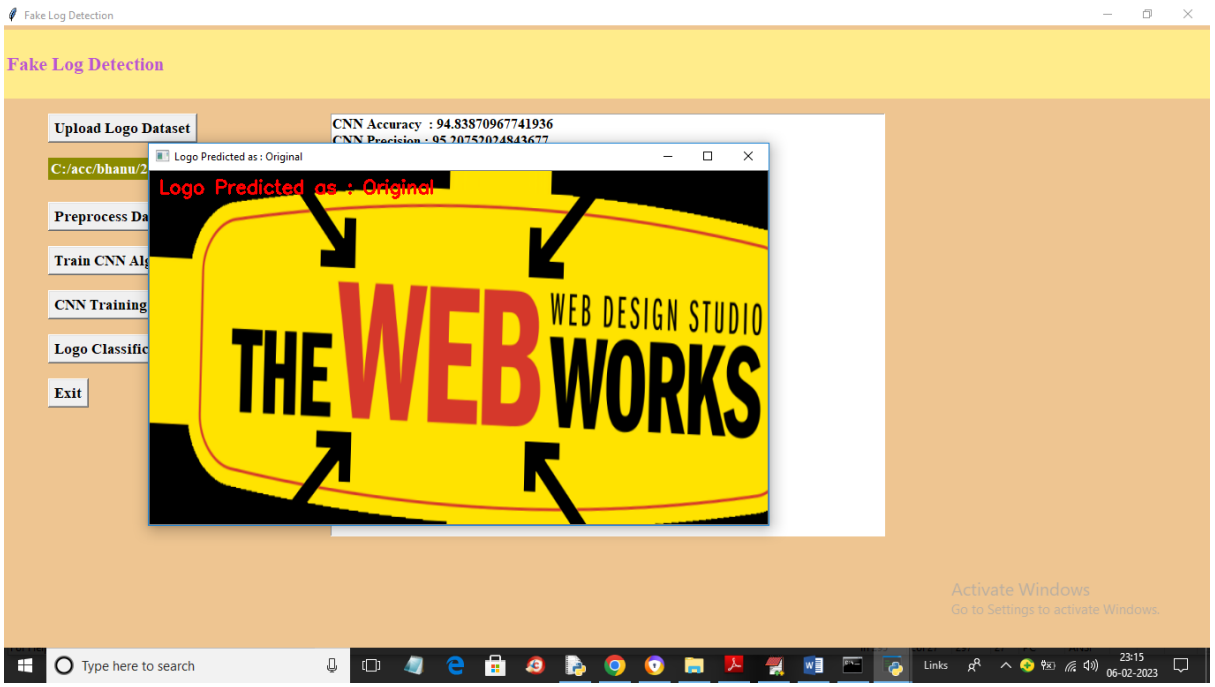

In above graph x-axis represents training epoch and y-axis represents Accuracy and Loss values and green line represents accuracy and red line represents loss and in above graph we can see with each increasing epoch accuracy got increase and loss got decrease. Now close above graph and then click on 'Logo Classification' button to get below output.




In above screen selecting and uploading logo image and then click on 'Open' button to get below output.




In above screen in red colour text we can see image logo classified as Fake and similarly you can upload and test other images.




Above logo classify as original logo image.

# 8. CONCLUSION

With logo detection, it’s easier to prevent fraud and keep a specific brand safe from counterfeit attempts. Adding this tool to a company’s protection strategy can help find fraudulent or harmful content involving a brand on any online platformthis Logo Detection app aims to help consumers distinguish forgeries from the original product. Using this system, a consumer can verify whether a product is in fact an original. This application can also be helpful for brands struggling to fight against forged products.

# 9 . FUTURE SCOPE

Lightweight logo detection: Lightweight networks are designed to reduce model complexity further while maintaining model accuracy. With the rapid development of mobile terminals such as mobile phones and computers, people pay more attention to lightweight detectors. Lightweight logo detection can help consumers quickly identify and accurately search for the products they need and thus can significantly improve the efficiency of online shopping. Although significant efforts have been made recently, the speed gap between the machine and the human eye is still huge. The detection accuracy is also insufficient, especially for small logos. Therefore, improving the accuracy of lightweight logo detectors is one of the future.

# 10. REFERENCES

[1]A.Smeulders,M. Worring, S. Santini, A. Gupta, and R. Jain, “Content based image retrieval at the end of the early years,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 12, pp. 1349–1380, Dec. 2000.

[2] R. Datta, D. Joshi, J. Li, and J. Z. Wang, “Image retrieval: Ideas, influences, and trends of the new age,” ACM Comput. Surv., vol. 40, No. 2, pp. 1–60, 2008.

[3] L. Ballan, M. Bertini, A. Del Bimbo, L. Seidenari, and G. Serra, “Event detection and recognition for semantic annotation of video,” Multimedia Tools Appl., vol. 51, no. 1, pp. 279–302, 2011.

[4] Y. Jing and S. Baluja, “Pagerank for product image search,” in Proc.WWW, Beijing, China, 2008, pp. 307–316.

[5] L. Ballan, M. Bertini, and A. Jain, “A system for automatic detection and recognition of advertising trademarks in sports videos,” in Proc. ACM Multimedia, Vancouver, BC, Canada, 2008, pp. 991–992.

[6] A. Watve and S. Sural, “Soccer video processing for the detection of advertisement billboards,” Pattern Recognit. Lett., vol. 29, no. 7, pp.994–1006, 2008.

[7] F. Leng, “A gradient balancing approach for robust logo detection,” in Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 4765–4769.

[8] W. Xu, Y. Liu, and D. Lin, “A simple and effective baseline for robust logo detection,” in Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 4784–4788.

[9] K. Tong, K. W. Cheung, and X. Yu, “ICME 2022 few-shot logo detection top 9 solution,” arXiv preprint arXiv:2206.11462, 2022.

[10] H. Sahbi, L. Ballan, G. Serra, and A. Bimbo, “Context-dependent logo matching and recognition,” IEEE Transactions on Image Processing, vol. 22, no. 3, pp. 1018–1031, 2013.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Communications of the ACM, vol. 60, no. 6, pp. 84 – 90, 2012.

[12] A. Joly and O. Buisson, “Logo retrieval with a contrario visual query expansion,” in Proceedings of the 17th ACM International Conference on Multimedia, 2009, pp. 581–584.

[13] S. Romberg, L. G. Pueyo, R. Lienhart, and V. Z. Roelof, “Scalable logo recognition in real-world images,” in Proceedings of the 1st ACM International Conference on Multimedia Retrieval, 2011, pp. 1–8.

[14] S. C. H. Hoi, X. Wu, H. Liu, Y. Wu, H. Wang, H. Xue, and Q. Wu, “Logo-Net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks,” arXiv preprint arXiv:1511.02462, 2015. [15] H. Su, X. Zhu, and S. Gong, “Open logo detection challenge,” in British Machine Vision Conference, 2018, pp. 111–119.

[16] Q. Hou, W. Min, J. Wang, S. Hou, Y. Zheng, and S. Jiang, “FoodLogoDet-1500: A dataset for large-scale food logo detection via multi-scale feature decoupling network,” in Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 4670–4679.

[17] J. Wang, W. Min, S. Hou, S. Ma, Y. Zheng, and S. Jiang, “LogoDet3K: A large-scale image dataset for logo detection,” ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 18, no. 1, pp. 1–19, 2022.

[18] C. Li, I. Feherv ´ ari, X. Zhao, I. Mac ´ edo, and S. Appalaraju, “SeeTek: ˆ Very large-scale open-set logo recognition with text-aware metric learning,” in IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 587–596.

[19] Y. Kalantidis, L. G. Pueyo, M. Trevisiol, V. Z. Roelof, and Y. Avrithis, “Scalable triangulation-based logo recognition,” in Proceedings of the 1st ACM International Conference on Multimedia Retrieval, 2011, pp. 1–7

[20] S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini, “Deep learning for logo recognition,” Neurocomputing, vol. 245, pp. 23–30, 2017.

[21] H. Su, X. Zhu, and S. Gong, “Deep learning logo detection with data expansion by synthesising context,” in IEEE Winter Conference on Applications of Computer Vision, 2017, pp. 530–539.

[22] Y. Liao, X. Lu, C. Zhang, Y. Wang, and Z. Tang, “Mutual enhancement for detection of multiple logos in sports videos,” in IEEE International Conference on Computer Vision, 2017, pp. 4856–4865.

[23] J. Liu, F. Shen, M. Wei, Y. Zhang, H. Zeng, J. Zhu, and C. Cai, “A large-scale benchmark for vehicle logo recognition,” in 4th International Conference on Image, Vision and Computing, 2019, pp. 479– 483.

[24] A. Kuznetsov and A. V. Savchenko, “A new sport teams logo dataset for detection tasks,” in International Conference on Computer Vision and Graphics, 2020, pp. 87–97.

# LOGO INFRINGEMENT DETECTION BY USING CNN

Shaik Karishma[1] , Y. Lakshmi Priya[2] , S.Supriya[3] , Dr.K.Lakshminadh[4]

[1, 2, 3]Student, Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Guntur (D.T) A.P, India

[4]Professor, Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Guntur (D.T) A.P, India

shaikkarishama782@gmail.com , priya9.yerramreddy@gmail.com ,

supriyasamudrala321@gmail.com , drklmn7@gmail.com

***Abstract*: Logo Infringement is mainly used for Logo Detection that means to check whether the logo is fake or original so we have to done this project by using CNN;the technologies are implemented in this project are Tensorflow and Keras these are most important technologies are implemented in this project.firstly we have to upload dataset of Logo after uploading the dataset we have to done Logo classification then we have to know whether the logo is fake or original.**

## INTRODUCTION

.

Logo Infringement Detection is used to identify the fake logo can be matching and recognized by original logo here every image that means Logo is divided into the rows and columns in this each cell can be represented as a pixels so the one image cell can be compared with the another image cell here the cell can be compared with the original logo thenit compared with the every original logo if it original logo keeps it with as original logo other wise it can be treated as fake. By using CNN algorithms we developed this project,the processing of the project first to upload a dataset that dataset have both fake and original logos while uploading the dataset we can observe the path of the file after uploading the dataset in preprocessing,preprocessing it means it tells the percentage of the test images and train images we actually have twenty percent of the train images and eighty percent of the test images after completing of the preprocessing to train the CNN algorithm based on that algorithm to display the CNN lost and gain percentage after completing of the training of the graph to classify the particular data,classification means to upload the any logo present in the dataset that display uploading logo is a fake or original.

original fake

## LITERATURE SURVEY

In Logo Infringement Detection we working with the conditions are like pattern of use , types of pictures ,semantics and the sensory gab,in this image or logo can be stored by color,texture and geometry.We also consider the similarity of the in image retrieval also consider the shape of the images ,signs this have interaction between the images

the ideas behind this project are developed by the some different types of techniques and systems.We also involved challenges in this project also discuss the swapping of the subfields and also many new people can involved here challenges are based on the technologies.

this project we can search for the images,while searching of the images we can search fake images and also original images and also to provide the best algorithms for group of images and idea behind this by using this retrieval of images we can verify the classification of the logos. s

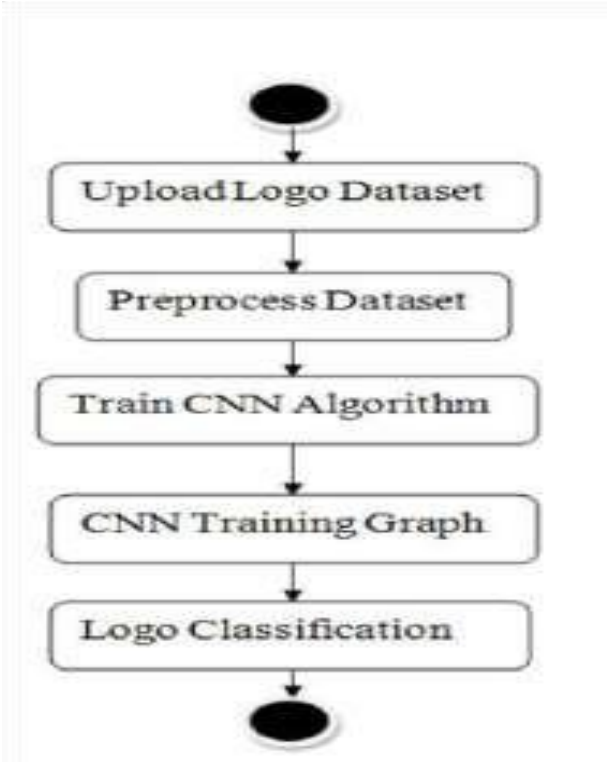
## System Analysis

Logo detection Accuracy of Existing system is low and the accuracy of the proposed system is high.In system analysis consists of steps that are present in the below.

**steps involved in this project are**

- Upload Logo Dataset
- Preprocess Dataset
- Train CNN Algorithm
- CNN Training Graph
- Logo classification

**FLOWCHART**




**UPLOAD LOGO DATASET**

Attributes are present in this Dataset are fake and original.After uploading the dataset it can display original and fake attributes andalso it can display the path of the file.

## preprocess Dataset

Preprocessing means to preparethe raw data.That raw data can be accept the network afte preprocessing,the dataset can display test andtrain of images.In the form of percentages . It contains 80% of images are train images and 20% ofthe images are test images

## Train CNN Algorithm

Training CNN Algorithm means to train thedataset.This dataset have fake and original logo. After training of CNN Algorithm itshows howmany fake and original logos are there.

## CNN Training Graph

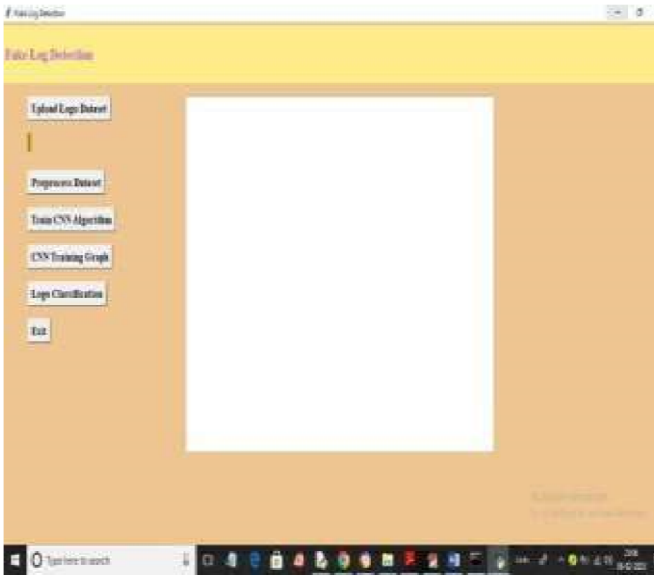
It can shows the accuracy of the dataset. Accuracy means quality,it shows loss and gain of the dataset.

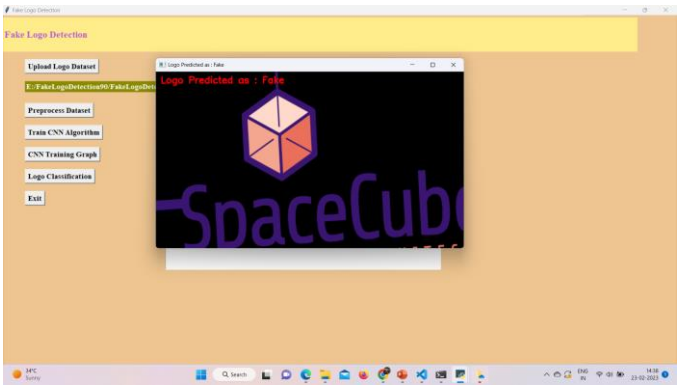
## Logo Classification

Logo classification classifies the logos and display whether the logo is fake or not ,these logos are taken from the dataset.

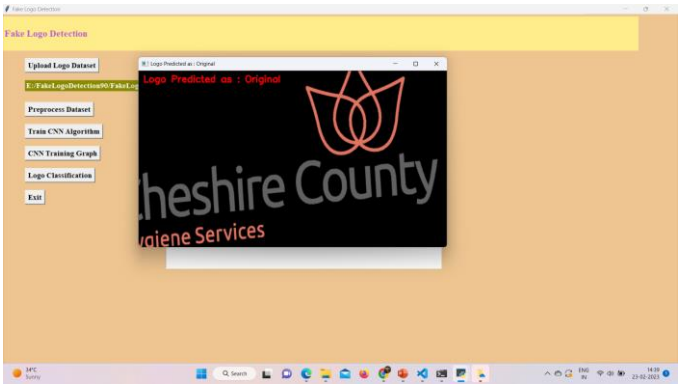
## Results

Upload Logo Dataset




Original Logo




## Conclusion

With logo detection, it's easier prevent fraud logos and keep a specific brand safe from counterfeit attempts. Adding this tool to a company's protection strategy can help find fraudulent or harmful content involving a brand on any online platform this Logo Detection app aims to help consumers.

## Referances

[1] A.Smeulders,M. Worring, S. Santini, A. Gupta, and R. Jain, "Content based image retrieval at the end of the early years," IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 12, pp. 1349–1380, Dec. 2000.

[2] R. Datta, D. Joshi, J. Li, and J. Z. Wang, “Image retrieval: Ideas, influences, and trends of the new age,” ACM Comput. Surv., vol. 40, No. 2, pp. 1–60, 2008.

[3] L. Ballan, M. Bertini, A. Del Bimbo, L. Seidenari, and G. Serra, “Event detection and recognition for semantic annotation of video,” Multimedia Tools Appl., vol. 51, no. 1, pp. 279–302, 2011.

[4] Y. Jing and S. Baluja, “Pagerank for product image search,” in Proc.WWW, Beijing, China, 2008, pp. 307–316.

[5] L. Ballan, M. Bertini, and A. Jain, “A system for automatic detection and recognition of advertising trademarks in sports videos,” in Proc. ACM Multimedia, Vancouver, BC, Canada, 2008, pp. 991–992.

[6] A. Watve and S. Sural, “Soccer video processing for the detection of advertisement billboards,” Pattern Recognit. Lett., vol. 29, no. 7, pp.994–1006, 2008.

[7] F. Leng, “A gradient balancing approach for robust logo detection,” in Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 4765–4769.

[8] W. Xu, Y. Liu, and D. Lin, “A simple and effective baseline for robust logo detection,” in Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 4784–4788.

[9] K. Tong, K. W. Cheung, and X. Yu, “ICME 2022 few-shot logo detection top 9 solution,” arXiv preprint arXiv:2206.11462, 2022.

[10] H. Sahbi, L. Ballan, G. Serra, and A. Bimbo, “Context-dependent logo matching and recognition,” IEEE Transactions on Image Processing, vol. 22, no. 3, pp. 1018–1031, 2013.

CG3

ORIGINALITY REPORT

| 6% | 6% | 6% | 6% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | www.ijera.com<br>Internet Source | 6% |

| | | | |
|---|---|---|---|
| Exclude quotes | On | Exclude matches | Off |
| Exclude bibliography | Off | | |


NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)
Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in
International Conference on
PAPER ID
NECICAIEA2K23076
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K23
17th & 18th March, 2023
Organized by Department of Computer Science and Engineering in Association with CSI
Certificate of Presentation
This is to Certify that Shaik Karishma , Narasaraopeta Engineering College has presented the paper title LOGO INFRINGEMENT DETECTION in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.
Convenor
Dr.S.V.N.Srinivasu
Chief-Convenor
Dr.S.N.Tirumala Rao
Principal, Patron
Dr. M. Sreenivasa Kumar
MHRD'S INNOVATION CELL (GOVERNMENT OF INDIA)


NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)
Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in
International Conference on
PAPER ID
NECICAIEA2K23076
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K23
17th & 18th March, 2023
Organized by Department of Computer Science and Engineering in Association with CSI
Certificate of Presentation
This is to Certify that Y.Lakshmi Priya , Narasaraopeta Engineering College has presented the paper title LOGO INFRINGEMENT DETECTION in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.
Convenor
Dr.S.V.N.Srinivasu
Chief-Convenor
Dr.S.N.Tirumala Rao
Principal, Patron
Dr. M. Sreenivasa Kumar


NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)
Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in
International Conference on
PAPER ID
NECICAIEA2K23076
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K23
17th & 18th March, 2023
Organized by Department of Computer Science and Engineering in Association with CSI
Certificate of Presentation
This is to Certify that S.Supriya , Narasaraopeta Engineering College has presented the paper title LOGO INFRINGEMENT DETECTION in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.
Convenor
Dr.S.V.N.Srinivasu
Chief-Convenor
Dr.S.N.Tirumala Rao
Principal, Patron
Dr. M. Sreenivasa Kumar
MHRD'S INNOVATION CELL (GOVERNMENT OF INDIA)


NARASARAOPETA
NEC ENGINEERING COLLEGE
(AUTONOMOUS)
Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in
International Conference on
PAPER ID
NECICAIEA2K23076
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K23
17th & 18th March, 2023
Organized by Department of Computer Science and Engineering in Association with CSI
Certificate of Presentation
This is to Certify that K.Lakshminadh , Narasaraopeta Engineering College has presented the paper title LOGO INFRINGEMENT DETECTION in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.
Convenor
Dr.S.V.N.Srinivasu
Chief-Convenor
Dr.S.N.Tirumala Rao
Principal, Patron
Dr. M. Sreenivasa Kumar