Mustansiriyah Journal of Pure and Applied Sciences

www.MJPAS.com

Comprehensive Image Classification using Hybrid CNN-LSTM Model with Advanced Feature Extraction on Coco Dataset

Zahraa Haimeed Rasool [1], Maha Adham Abdel Amir [2]

[1,2] Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, Iraq.

ABSTRACT

A fundamental computer vision challenge is object detection, which involves pinpointing and classifying objects in an image or video. This capability opens up many possibilities in autonomous vehicles, surveillance, and image analytics. In this study, the proposed hybrid CNN_LSTM model is employed to classify various categories within the Coco dataset, spanning common everyday objects, animals, vehicles, and more. The workflow includes steps to enhance image data and extract pertinent features. Initially, RGB images were converted to grayscale to simplify processing, followed by histogram equalization to enhance the contrast and median blur for noise reduction. Principal Component Analysis (PCA), Gray-Level Co-Occurrence Matrix (GLCM), and Histogram of Oriented Gradients (HOG) were used for feature extraction. The architecture employs a proposed hybrid CNN_LSTM model structure, combining Convolutional Neural Spatial and sequential patterns are captured by CNNs and LSTM networks. This effective hybrid neural network classifies images using preprocessing and feature extraction. The model performed well on the COCO dataset, with an accuracy of 0.9917, precision of 0.991738, recall of 0.991695, and F1 score of 0.999949, supported by consistent loss reduction and accuracy improvement in its training history, proving its pattern recognition abilities.

***Keywords*:** Object Detection, Principal Component Analysis, Gray-Level Co-Occurrence Matrix, Histogram of Oriented Gradients, Convolutional Neural Networks, Long Short-Term Memory.

1. Introduction

The significance of content-based annotation in object image recognition is critical for monitoring applications. With the rapid advancements in remote sensing technologies, the abundance of data generated by advanced objects, such as satellite imagery, presents valuable resources for a multitude of real-world applications, from urban planning to ecological monitoring [1]. The vastness of this data repository presents both opportunities and challenges. In this study, we focus on label image categorization, aiming to assign remote sensing images to a set of predefined object-level tags. Automated annotation not only

*Corresponding author : Zahraa Haimeed Rasool
E-mail address:

expedites image retrieval but also provides semantic concepts for diverse applications. Label scene recognition is common in computer vision applications[2]. In real-world scenarios, images can encapsulate a variety of correlated tags and convey objects within a region of interest. Although deep learning methods have transformed image processing and computer vision, their application in object image recognition remains nascent, particularly because of the scarcity of labeled datasets[3]. This study addresses this gap by aiming to enhance label recognition. However, these challenges persist in the future. Extracting object-level features for label tasks remains complex, given the variation in label visibility across images. Advances in label dependencies, such as sequence- and graph-based approaches, have emerged to enhance recognition capabilities[4]. Reliable labels in a sequence or graph can serve as predictive information for the other labels. Notably, the distribution of concepts across modalities (image, graph, and text) presents opportunities for cross-modal feature enhancement. Object detection is intended to locate and classify an object within a specific picture or frame in video format. This objective is accomplished by merging several complicated operations, such as object localization and classification. Different methodologies have been used to address these problems. Nevertheless, progress in Deep Learning, including neural network architectures, has opened up a new epoch of Object Detection, where convolutional neural networks (CNNs) have become an industry's standard for image classification tasks; meanwhile, Recurrent Neural Networks (RNN The current paper presents a novel CNN-LSTM hybrid architecture, which capitalizes on the complementary strengths of both paradigms, and offers an all-inclusive response to image classification problems.

One of the main advantages of this study is that it proposes and assesses an object detection and classification model based on a hybrid CNN-LSTM architecture. CNNs have proven very effective in extracting spatial features from images, but mixing LSTM networks allows for capturing consecutive patterns that are imperative to object detection tasks. This study proposes a hybrid model architecture that is expected to exploit the advantages associated with each of these neural network types. Its intention is, therefore, to increase the reliability and validity of object categorization and thus progress to present-day computer vision [5].

This study adds to this contribution as it explores the crucial issues of preprocessing and feature extraction. For example, depending on the quality of the input data, many other object detection tasks are affected by the performance of their models. Therefore, the authors proposed a set of image data preprocessing techniques and input the processed data into the hybrid CNN-LSTM model. Techniques such as converting RGB images into grayscale, enhancing the contrast using histogram equalization, and using median blur for noise reduction have been used. Furthermore, this study covers modern feature identification techniques such as Principal Component Analysis (PCA), Gray-Level Co-Occurrence Matrix (GLCM), and Histogram of Oriented Gradients (HOG). Collectively, such techniques add to the dataset information that helps the classification model become better informed and, thus, make fewer wrong conclusions.

Another important aspect of this study’s contribution lies in the selection of the dataset for evaluation. The Coco dataset constitutes a commonly used benchmark for the computer vision community and includes various object classes that are usually found. In addition to demonstrating the effectiveness of the Hybrid CNN-LSTM model by testing it on the Coco dataset, this paper highlights its adaptability with regard to many different types of objects, including ordinary household objects, animals, automobiles, and others. The proposed model is applicable to various other settings, where multiple objects belonging to different categories must be recognized in one instance..

1. **Related Work**

Previous research, preprocessing, feature extraction, and classification methods are covered in this section. The object category research community is increasingly interested in deep learning-based methods. Hong et al. (2019) [6] investigated deep learning-based bird detection using UAV imagery. Their work uses sensor data and deep learning algorithms to improve bird detection in unmanned aerial vehicles. This shows how deep learning improves bird detection strategies for various applications. Willi et al. (2019) [7] connected citizen science and deep learning to identify animals. This Methods in Ecology and Evolution study classifies species from camera trap images. The authors used CNNs to enable automatic species recognition and evaluate non-expert data labeling. This study illuminates species identification through technology and community engagement. Yudin et al. (2019) [8] contribute Deep Learning-based large animal detection in road object images. They used CNNs for feature extraction and classification. Their data-augmented approach detected large animals in road scenes, providing insights into deep learning wildlife detection challenges. Li and Peng (2022) [9] present a cross-modal feature learning, label graph mining, and Residual Multi-Attentional CNN LSTM network-based aerial object classification method. This architectural innovation captures the complex relationships between visual content and labels, improving classification accuracy. Label graph mining helps understand label interdependencies, and experimental validation shows its superior performance. Hussan et al. (2022) [10] presented real-time object detection and recognition for the blind. The proposed system uses deep-learning for object detection and recognition to help the visually impaired in real time. This approach was tested for feasibility and efficacy, confirming its potential benefits for visually impaired people. Saurav et al.(2022)[11] provided a dual integrated convolutional neural network (DICNN) model for real-time facial expression recognition in difficult conditions. The DICNN model balances recognition accuracy and computational efficiency with 1.08M parameters and 5.40 MB memory storage, achieving competitive accuracy and significantly improved execution speed on a resource-constrained embedded platform. Bi (2022) [12], introduces a keyframe-based violent behavior recognition method that treats video frames as independent events, reduces hardware requirements, and addresses interference. This scheme performs better than existing methods on multiple datasets, achieving state-of-the-art violence recognition. Murugesan, et al(2022) [13], presents an efficient hybrid deep learning model for lung nodule segmentation and classification from CT images to detect lung cancer early. By using an adaptive median filter to reduce noise and a U-net architecture for segmentation, the model identified and classified lung cancer using image analysis. Saroja et al. (2023)[14] presented an image-captioning method that outperformed existing methods by 11-15 percent using an improved YOLO V5 model for object detection and an Xception V3 model for caption generation. This method produces voice and text captions in multiple languages for visually impaired people.

This study notably combines deep learning and advanced feature extraction methods for picture classification, unlike earlier works on deep learning in object detection. A Hybrid CNN-LSTM model with advanced feature extraction is applied to the Coco Dataset for the first time. The proposed design enhances classification accuracy and captures complex visual content-label interactions. An innovative image-captioning method using state-of-the-art models helps visually impaired people, demonstrating the study's unique and impactful contributions to the field.

2. Deep Learning in Object Detection

Object detection has improved in terms of both accuracy and speed owing to deep learning, which has drastically transformed this field. Object detection plays a key role in computer vision that finds its uses across different areas, such as self-driving cars, surveillance systems, and medical imaging. Presently, modern object detectors are based on deep learning techniques, especially CNNs, including other architectures. This section focuses on exploring the concepts, progress, and use of object detection based on deep learning.

3.3.1 Convolutional Neural Network Layer and Architecture

Convolutional Neural Networks (CNN) have four key layers: convolution, pooling, fully connected, and nonlinear layers(see figure 1). Each layer has a specific function and is crucial for the operation of the CNN, as shown below:

1. The convolutional Layer (Eq. 1) The first layer, convolution, involves applying learnable filters to the input image. These filters slide over the image, compute dot products, and apply a nonlinear activation function (e.g., ReLU). The results are called feature maps, and they stack to form an output volume. Parameters such as depth, stride, and padding influence the output size[15].

$$Fi = f(\,W.V_{i:i+h-1} + b) \tag{1}$$

where ***Fi*** is the vector of features produced from a convolutional operation and W *is* the filter used by identified with window size (h).$\boldsymbol{V_{i:j+h-1}}$: the local vector from position *i* to position *i:i+h-1* in the vector v, *b* equal to biased. ***f*** : non-linear hyperbolic tangent

2. Pooling Layer: After convolution, the pooling layer reduces the feature dimensionality and enhances the model invariance to variations such as rotation and scale. This helps to improve generalization and mitigate overfitting [16].
3. Fully Connected Layer: In layer, classification is performed by applying weights to features extracted from the previous layers. The Softmax function determines class probabilities. The number of fully connected layers and neurons affects CNN performance[17].

$$y_{jk}(x)=f(\sum_{i=1}^{nH} w_{jk}\ x_i + w_{j0} \ldots\ldots\ldots\ldots\ (2)$$

where the weight matrix W and the input vector x. The bias term (W0) can be added to the nonlinear function.

4. Nonlinearity Layer: Nonlinear activation functions are applied to neuron inputs, allowing CNNs to learn complex patterns. Activation functions are essential for error backpropagation and for enhancing discrimination ability [18].By capitalizing on the representational potential, the utilization of polynomial equations in the convolutional operator is suggested, thereby incorporating non-linear convolutions.

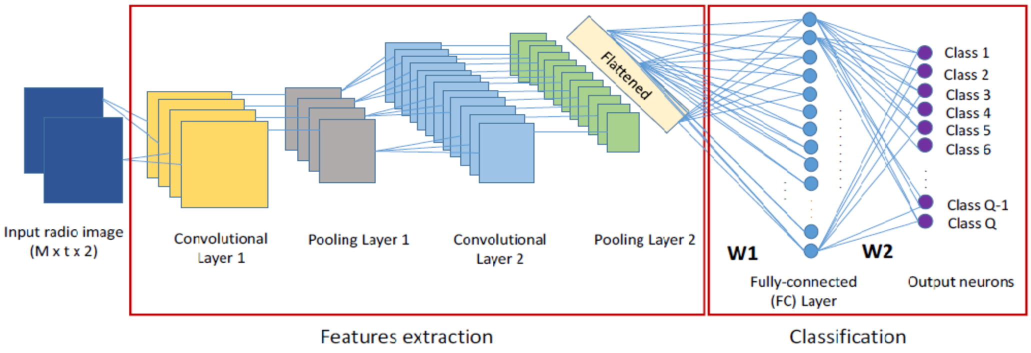


Figure 1. CNN architecture using two convolutional layers and a fully connected layer[19].

3.3.2 Deep Learning Long Short-Term Memory (LSTM) Models

Deep learning models, specifically LSTM networks within the RNN framework, are adept at handling sequential data such as network traffic analysis and gaining recognition for their pattern recognition capabilities, with the aim of improving accuracy, reducing false alarms, and addressing class imbalances [20]. LSTMs overcome RNN limitations, notably the vanishing gradient problem, by using memory cells and gating mechanisms. They excel in modeling sequential data, preserving long-term dependencies, and mitigating gradient vanishing [21]. Figure 2 illustrates the LSTM architecture, comprising distinct layers with sigmoid and hyperbolic tangent activation functions, including the input, forget, output gates, memory cells, and hidden states [22].

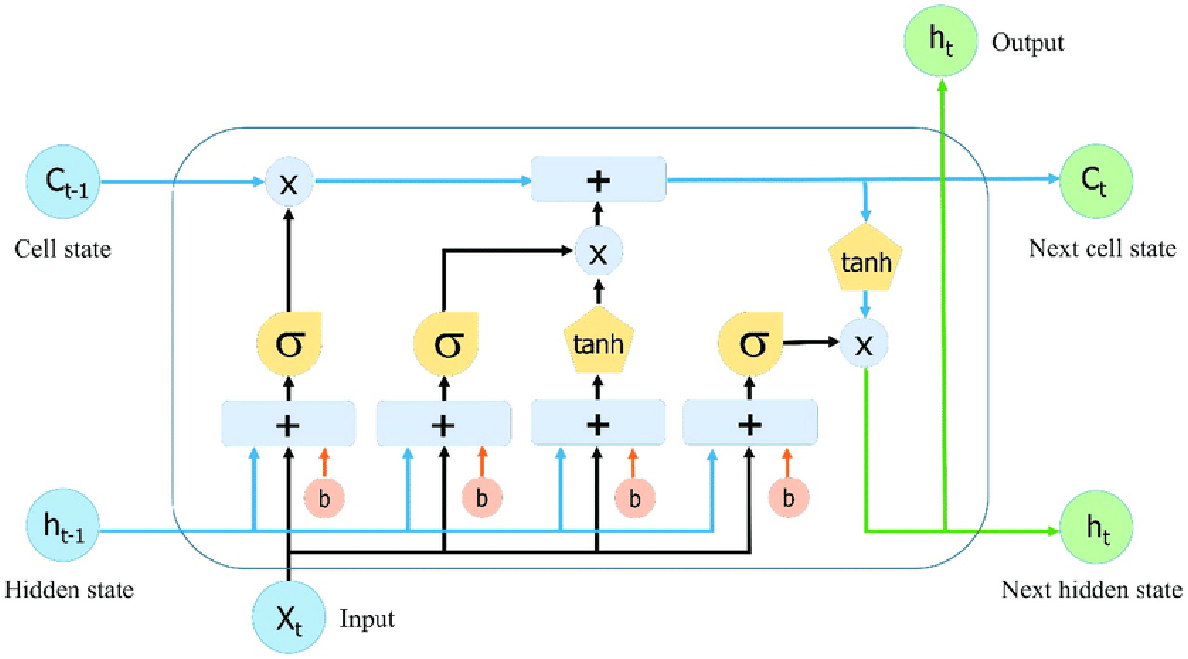


Figure 2. Simple structure of the LSTM architecture network [23].

3. **Methodology**

This section outlines the comprehensive approach undertaken to develop and assess a hybrid CNN-LSTM model for complete classification purposes utilizing the Coco dataset. The methodology encompasses several key stages, namely data preprocessing, feature extraction techniques, development of the model architecture, and implementation of an effective training strategy.

3.1 Dataset

The Common Objects in Context (COCO) [24] is Microsoft-sponsored image recognition, segmentation, and captioning dataset. This dataset's open source has made great strides in semantic segmentation and become a "standard" for image semantic understanding, but COCO faces its own challenge. 330k images, 1.5 object instances, and 5 captions per image make up this large dataset. The literature uses COCO extensively. The dataset's non-iconic images make it ideal for image captioning. Unlike non-iconic images, iconic images have one object with a background. Images are labeled carefully to account for object layout, which helps establish scene context. Image count is 526394. Distributed over 92 classes and split 70/30 between Train and Test. Testing has 157,918.2 images and training 368,475.8.

4.1 The Prepressing Phase

Preprocessing techniques optimize raw data by performing noise reduction, normalization, and data cleaning to enhance data quality for analysis. In this section, we describe the conversion to grayscale, Histogram equalization, and Median Blur [25].

1. Conversion to Grayscale:

The conversion of RGB photos to grayscale reduces their color information to one intensity channel, making processing and analysis easier. To calculate the grayscale pixel value for each pixel in the RGB image, calculate the weighted sum of the red, green, and blue components [26]. The formula for this conversion is given by Eq. (1) can be expressed as follows:.

$$\text{Grayscale (i,j)} = (0.2989*r) + (0.5878*g) + (0.1140*b) \qquad (1)$$

Here, IRGB represents the original RGB image and IGray represents the resulting grayscale representation. Coefficients 0.2989, 0.5870, and 0.1140 are weights that consider the importance of each color channel in human perception [27].

2. Histogram Equalization

Histogram equalization is a technique that enhances image contrast by redistributing pixel intensities. Given an image with histogram h(i), the transformation function T(i) is formulated as Eq. (2) [28].

$$\text{Hist(v)} = \text{cut}\left(\frac{(cdf_{(v)} - cdf_{min}}{(m*n)-1}\right) * (L\text{-}1) \qquad (2)$$

Where Cdf represents the cumulative distribution function, (m,n) represents the image's node of pixels, and L represents the number of grey levels used (256) [29].

3. Median Blur:

Image noise is reduced using median blur while edges and fine features are preserved. It replaces each pixel's value with its neighborhood median. A sliding window or kernel of size (2n+1)×(2n+1) defines the neighborhood, with n determining the local area's extent. Eq. (3) calculates the output pixel at coordinates (x,y) within kernel K centered at (x, y) [30] :

$$O(x, y) = \text{Median}(K(x - k, y - k), K(x - k, y - k + 1), ..., K(x + k, y + k - 1), K(x + k, y + k)) \ldots (3)$$

4.2 Feature Extraction

Feature extraction in data analysis involves the extraction of essential information from raw data to create a concise feature set, reduce dimensionality, and capture critical patterns for analysis. In this section description the Principal Component Analysis (PCA), Gray-Level Co-Occurrence Matrix (GLCM) , and Histogram of Oriented Gradients [31].

1. Principal Component Analysis (PCA):

PCA reduces dimensionality by keeping the most variance in a high-dimensional dataset [32], [33]. This is done by detecting and capturing the data's primary components, which are linear combinations of starting features. To describe this analytically, we use a centered dataset matrix X_bar, which represents the data mean. First, calculate the covariance matrix C using this formula:

$$C=1/n\ (\textstyle\sum_{i=1}^{n}(\mathrm{x}_i-\bar{\mathrm{x}})(\mathrm{x}_i-\bar{\mathrm{x}})^T \qquad (4)$$

where, n is the number of observations. x_i is a data point. $\bar{x}$ is the mean of the data.

2. Gray-Level Co-Occurrence Matrix (GLCM):

The Gray-Level Co-Occurrence Matrix (GLCM) is a crucial texture analysis approach for assessing the prevalence of pixel pairings with different intensity fluctuations at different orientations in a picture. Its importance is on capturing images' textural details [34]. For an image with discrete grey levels, the GLCM (P) calculates the probability of encountering two pixels with given intensity values at a defined spatial offset (δx, δy).

$$P(i,j,\delta,\theta)=1/N\ \textstyle\sum_{i=1}^{w-\delta}\sum_{j=1}^{H}\delta(x,y)\delta(x+\delta,y)\ \ldots\ldots(5)$$

In this context, (δx, δy, θ) represents the normalized Gray-Level Co-occurrence Matrix (GLCM) at a specific spatial offset ((δx, δy)) and direction (θ). P(δx, δy, θ) corresponds to a specific element in the GLCM, denoted as P, for an identical offset and direction. Ng signifies the number of grey levels present in an image[35].

3. Histogram of Oriented Gradients

The Histogram of Oriented Gradients (HOG) is a common computer vision method for object detection that computes gradient magnitudes and orientations within image cells, builds orientation histograms, normalises them within blocks, and merges them to create a feature vector that describes the object's contours and surface attributes [36].

$$\begin{array}{l} G\_x = I * K\_x \\ G\_y = I * K\_y \\ \text{magnitude} = \text{sqrt}(G\_x\text{^}2 + G\_y\text{^}2) \\ \text{orientation} = \text{arctan2}(G\_y, G\_x) \end{array} \qquad (6)$$

where includes the gradient components (G_x and G_y), image (I), gradient kernels (K_x and K_y), gradient magnitude (magnitude), and gradient orientation (orientation). Although the precise equations and implementation details can be intricate, HOG simplifies object recognition by encapsulating localized patterns of gradient orientations [37].

4.3 Deep Learning Classification

Deep learning uses CNNs and RNNs to learn hierarchical features for accurate data classification [38]. This architecture relies on the Hybrid CNN-LSTM structure, which combines CNNs and LSTMs. This novel approach aims to capture the intricate relationship between spatial and sequential data patterns. Convolutional Layers (conv_1d), Max Pooling Layers (max_pool), and LSTM Layers (lstm_1) formed the network architecture.

Synergistically and sequentially, these layers process and extract data features. This architecture's peak is a Dense Layer with 80 units (dense1) as shown in table4.

Table 1. The Architecture proposed hybrid CNN_LSTM model Layers Detailed Information.

Type	Filter	Parameter
conv1d_1 (Conv1D)	(None, 396, 16)	96
max_pooling1d_1	(MaxPooling1 (None, 392, 32)	0
conv1d_2 (Conv1D)	(None, 392, 32)	2592
conv1d_2 (Conv1D)	(None, 392, 32)	0
lstm_1 (LSTM)	(None, 392, 32)	8320
max_pooling1d_3	(MaxPooling1 (None, 392, 32)	0
conv1d_3 (Conv1D)	(None, 388, 32)	5152
max_pooling1d_4	(MaxPooling1 (None, 388, 32)	0
lstm_2 (LSTM)	(None, 388, 16)	3136
max_pooling1d_5	(MaxPooling1 (None, 388, 16)	0
flatten_1 (Flatten)	(None, 6208)	0
dense_1 (Dense)	(None, 92	571228
Total PARAM's: 590,524, Trainable PARAM's: 590,524, Non-trainable PARAM's: 0		

Training and adapting the neural network architecture's 590,524 parameters is possible. This sophisticated design highlights the network's complicated image-categorization capabilities. It uses the hybrid CNN_LSTM model to classify objects, animals, cars, and more in the Coco dataset. This method follows a carefully planned sequence to improve image data and extract relevant information. This method successfully categorizes complex images using preprocessing, feature extraction, and a hybrid neural network design. This methodology emphasizes seamless integration of preprocessing, feature extraction, and hybrid neural network architecture to solve complex image categorization problems (see algorithm (1)).

Algorithm (1): The Methodology of Proposed Work

Input Data: The Coco dataset.

Output: Classification and Analysis of the Coco Dataset Using the Trained Hybrid CNN-LSTM Model Classifying Different Classes.

Step 1: Data Preparation

- Preparation for getting the Coco dataset.

Step 2: Image Preprocessing

- Convert Images: Eq (1) converts RGB images into grey scale for ease of further process.
- Implement Histogram Equalization: Optimize feature extraction by enhancing image contrast using Eq(2) .

- Perform Median Blur Operation: applying the median blur operation using Eq(3) .

Step 3: Feature Extraction

- Apply Principal Component Analysis (PCA): Reduce dimensionality using Eq(4) to find out essential data components.
- Compute Gray-Level Co-Occurrence Matrix (GLCM): provide us with texture features and spatial dependencies using Eq(5) .
- Utilize Histogram of Oriented Gradients (HOG): Compute gradient orientations for edge and shape detection (Eq 6).

Step 4: Model Architecture

- Built hybrid CNN-LSTM architecture that detects both spatial and temporal features in the dataset.

Step 5: Training and Evaluation

- Create train and test sets of data.
- The hybrid CNN-LSTM is trained on the learning data.

Step 6: Results and Analysis

- Measure the overall performance of the model on test data, using metrics like accuracy, confusion matrix etc.
- Evaluation of Coco Dataset Classes Classification Model.

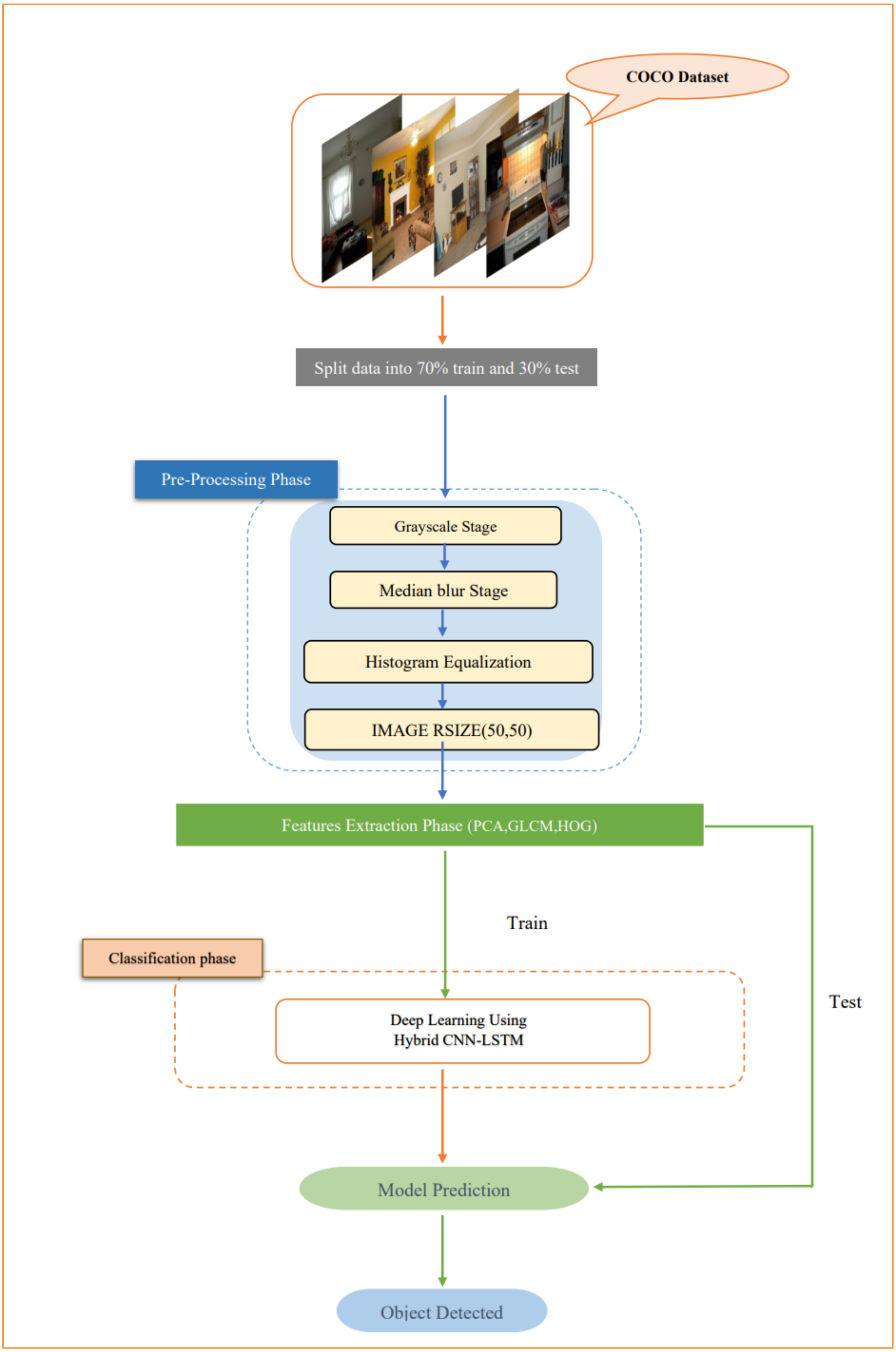

COCO Dataset
Split data into 70% train and 30% test
Pre-Processing Phase
Grayscale Stage
Median blur Stage
Histogram Equalization
IMAGE RSIZE(50,50)
Features Extraction Phase (PCA,GLCM,HOG)
Train
Classification phase
Deep Learning Using Hybrid CNN-LSTM
Test
Model Prediction
Object Detected

Figure 3. The proposed hybrid CNN_LSTM model flowchart

4 . Result and Discussion

In this section, we present the results and analysis of the proposed hybrid CNN-LSTM approach for classifying various categories within the Coco dataset using Python on the Google Colab platform. It utilizes the versatile Coco dataset, renowned for its extensive image variety, diverse properties (resolutions, RGB, grayscale), and a wide range of classes, enabling comprehensive evaluation of the proposed methodologies. the COCO training validation and test sets containing more 200000 images and 250 person instance labeled shown in figure 4.

Figure 4. Sample images of COCO dataset

To initiate this process, RGB images from the dataset underwent a series of preprocessing steps. First, they are converted to grayscale to simplify the subsequent processing steps by reducing the image from three color channels (RGB) to a single channel (gray). Subsequently, histogram equalization was applied to enhance the image contrast, rendering them more suitable for effective feature extraction. To further enhance the image quality, a median blur operation is employed to mitigate noise, ensuring that the subsequent stages of analysis are conducted on clean and processed data, as shown in figure 5.

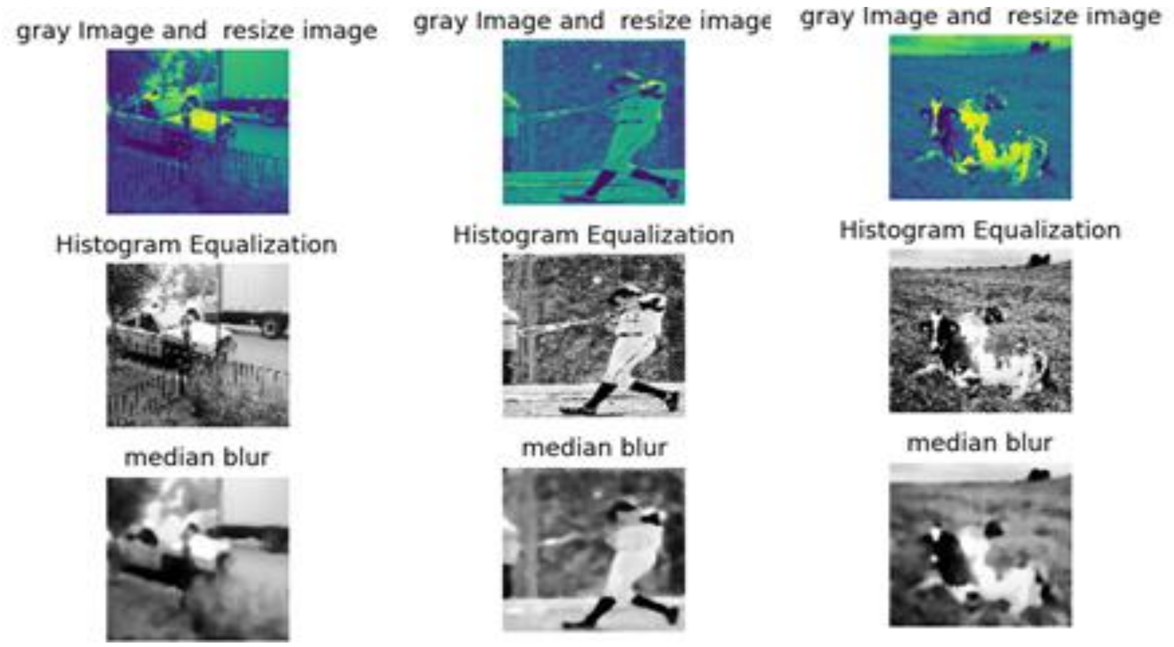
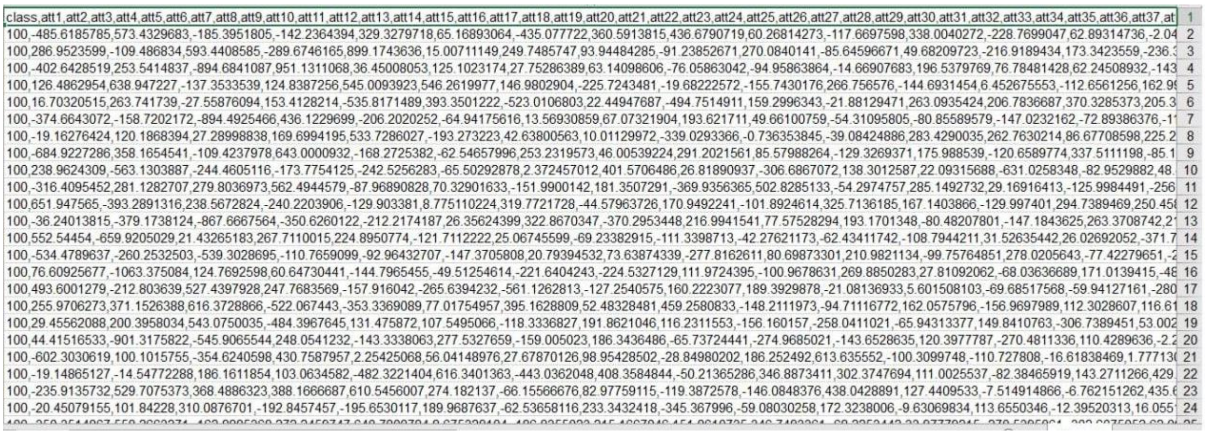


Figure 5. show the preprocessing Sample images of COCO dataset

The subsequent stages are dedicated to feature extraction, in which crucial data attributes are extracted. Principal Component Analysis (PCA) is applied to identify and retain essential data components while reducing dimensionality. A Gray-Level Co-Occurrence Matrix (GLCM) was computed to capture texture features and spatial relationships within the images. Additionally, the Histogram of Oriented Gradients (HOG) method is enlisted to detect object edges and shapes by analyzing the distribution of gradient orientations in the images as shown in Tables (2),(3) , and (4)

Table 2: Sample features after applying principal component analysis to the COCO dataset.



```
class,att1,att2,att3,att4,att5,att6,att7,att8,att9,att10,att11,att12,att13,att14,att15,att16,att17,att18,att19,att20,att21,att22,att23,att24,att25,att26,att27,att28,att29,att30,att31,att32,att33,att34,att35,att36,att37,at
100,-485.6185785,573.4329683,-185.3951805,-142.2364394,329.3279718,65.16893064,-435.077722,360.5913815,436.6790719,60.26814273,-117.6697598,338.0040272,-228.7699047,62.89314736,-2.04
100,286.9523599,-109.486834,593.4408585,-289.6746165,899.1743636,15.00711149,249.7485747,93.94484285,-91.23852671,270.0840141,-85.64596671,49.68209723,-216.9189434,173.3423559,-236.
100,-402.6428519,253.5414837,-894.6841087,951.1311068,36.45008053,125.1023174,27.75286389,63.14098606,-76.05863042,-94.95863864,-14.66907683,196.5379769,76.78481428,62.24508932,-143
100,126.4862954,638.947227,-137.3533539,124.8387256,545.0093923,546.2619977,146.9802904,-225.7243481,-19.68222572,-155.7430176,266.756576,-144.6931454,6.452675553,-112.6561256,162.9
100,16.70320515,263.741739,-27.55876094,153.4128214,-535.8171489,393.3501222,-523.0106803,22.44947687,-494.7514911,159.2996343,-21.88129471,263.0935424,206.7836687,370.3285373,205.3
100,-374.6643072,-158.7202172,-894.4925466,436.1229699,-206.2020252,-64.94175616,13.56930859,67.07321904,193.621711,49.66100759,-54.31095805,-80.85589579,-147.0232162,-72.89386376,-1
100,-19.16276424,120.1868394,27.28998838,169.6994195,533.7286027,-193.273223,42.63800563,10.01129972,-339.0293366,-0.736353845,-39.08424886,283.4290035,262.7630214,86.67708598,225.2
100,-684.9227286,358.1654541,-109.4237978,643.0000932,-168.2725382,-62.54657996,253.2319573,46.00539224,291.2021561,85.57988264,-129.3269371,175.988539,-120.6589774,337.5111198,-85.1
100,238.9624309,-563.1303887,-244.4605116,-173.7754125,-242.5256283,-65.50292878,2.372457012,401.5706486,26.81890937,-306.6867072,138.3012587,22.09315688,-631.0258348,-82.9529882,48.
100,-316.4095452,281.1282707,279.8036973,562.4944579,-87.96890828,70.32901633,-151.9900142,181.3507291,-369.9356365,502.8285133,-54.2974757,285.1492732,29.16916413,-125.9984491,-256
100,651.947565,-393.2891316,238.5672824,-240.2203906,-129.903381,8.775110224,319.7721728,-44.57963726,170.9492241,-101.8924614,325.7136185,167.1403866,-129.997401,294.7389469,250.45
100,-36.24013815,-379.1738124,-867.6667564,-350.6260122,-212.2174187,26.35624399,322.8670347,-370.2953448,216.9941541,77.57528294,193.1701348,-80.48207801,-147.1843625,263.3708742,2
100,552.54454,-659.9205029,21.43265183,267.7110015,224.8950774,-121.7112222,25.06745599,-69.23382915,-111.3398713,-42.27621173,-62.43411742,-108.7944211,31.52635442,26.02692052,-371.7
100,-534.4789637,-260.2532503,-539.3028695,-110.7659099,-92.96432707,-147.3705808,20.79394532,73.63874339,-277.8162611,80.69873301,210.9821134,-99.75764851,278.0205643,-77.42279651,-2
100,76.60925677,-1063.375084,124.7692598,60.64730441,-144.7965455,-49.51254614,-221.6404243,-224.5327129,111.9724395,-100.9678631,269.8850283,27.81092062,-68.03636689,171.0139415,-48
100,493.6001279,-212.803639,527.4397928,247.7683569,-157.916042,-265.6394232,-561.1262813,-127.2540575,160.2223077,189.3929878,-21.08136933,5.601508103,-69.68517568,-59.94127161,-280
100,255.9706273,371.1526388,616.3728866,-522.067443,-353.3369089,77.01754957,395.1628809,52.48328481,459.2580833,-148.2111973,-94.71116772,162.0575796,-156.9697989,112.3028607,116.61
100,29.45562088,200.3958034,543.0750035,-484.3967645,131.475872,107.5495066,-118.3336827,191.8621046,116.2311553,-156.160157,-258.0411021,-65.94313377,149.8410763,-306.7389451,53.002
100,44.41516533,-901.3175822,-545.9065544,248.0541232,-143.3338063,277.5327659,-159.005023,186.3436486,-65.73724441,-274.9685021,-143.6528635,120.3977787,-270.4811336,110.4289636,-2.2
100,-602.3030619,100.1015755,-354.6240598,430.7587957,2.25425068,56.04148976,27.67870126,98.95428502,-28.84980202,186.252492,613.635552,-100.3099748,-110.727808,-16.61838469,1.77713
100,-19.14865127,-14.54772288,186.1611854,103.0634582,-482.3221404,616.3401363,-443.0362048,408.3584844,-50.21365286,346.8873411,302.3747694,111.0025537,-82.38465919,143.2711266,429.
100,-235.9135732,529.7075373,368.4886323,388.1666687,610.5456007,274.182137,-66.15566676,82.97759115,-119.3872578,-146.0848376,438.0428891,127.4409533,-7.514914866,-6.762151262,435.
100,-20.45079155,101.84228,310.0876701,-192.8457457,-195.6530117,189.9687637,-62.53658116,233.3432418,-345.367996,-59.08030258,172.3238006,-9.63069834,113.6550346,-12.39520313,16.055
```

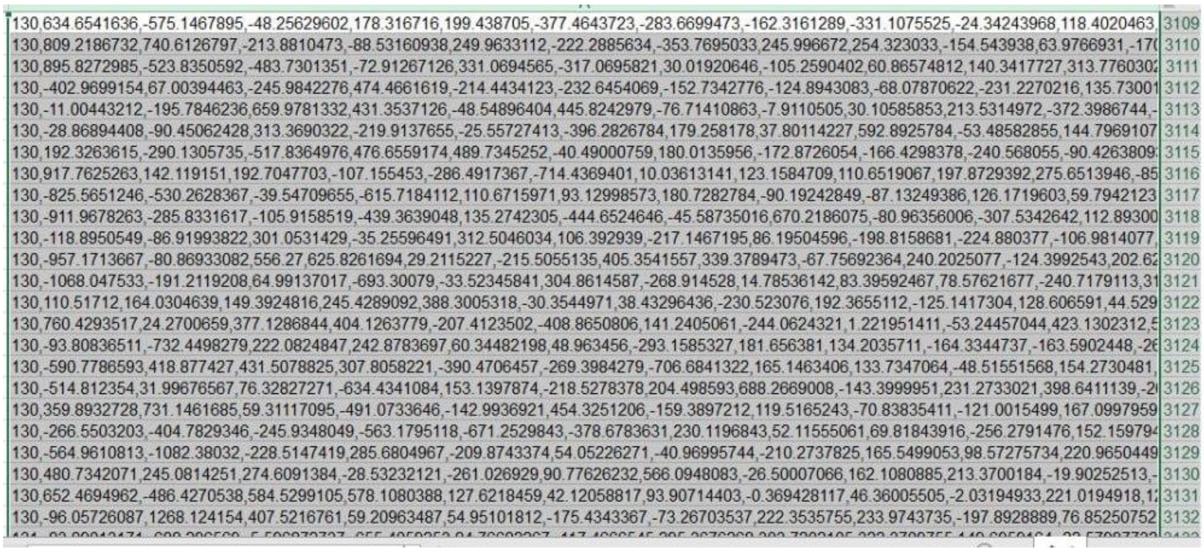
Table 2 provides a numerical representation of how each image is decomposed into various PCA components ranging from PCA_0 to PCA_2559, allowing for a more compact and informative representation of the image data.

Table 3. Sample features After Applying GLCM on the COCO dataset.

Image Name	Contrast	Correlation	…	Energy	Homogeneity
000000066584.jpg	248.627	0.912331	…	0.017706	0.15107
000000437301.jpg	218.511	0.961441	…	0.014928	0.198451
000000325813.jpg	476.2354	0.94186	…	0.015802	0.19307
000000055191.jpg	480.3186	0.873634	…	0.012958	0.103904
…	…	…	…	…	…
000000438678.jpg	250.8567	0.959828	…	0.022314	0.240668

Table 3 displays the computed GLCM features, such as Contrast, Correlation, Energy, and Homogeneity for diverse dataset images, collectively contributing to the characterization of texture and spatial interrelationships within the images.

Table 4. Sample features After Applying HOG on the COCO dataset.



```
130,634.6541636,-575.1467895,-48.25629602,178.316716,199.438705,-377.4643723,-283.6699473,-162.3161289,-331.1075525,-24.34243968,118.4020463,  3109
130,809.2186732,740.6126797,-213.8810473,-88.53160938,249.9633112,-222.2885634,-353.7695033,245.996672,254.323033,-154.543938,63.9766931,-17  3110
130,895.8272985,-523.8350592,-483.7301351,-72.91267126,331.0694565,-317.0695821,30.01920646,-105.2590402,60.86574812,140.3417727,313.776030  3111
130,-402.9699154,67.00394463,-245.9842276,474.4661619,-214.4434123,-232.6454069,-152.7342776,-124.8943083,-68.07870622,-231.2270216,135.73001  3112
130,-11.00443212,-195.7846236,659.9781332,431.3537126,-48.54896404,445.8242979,-76.71410863,-7.9110505,30.10585853,213.5314972,-372.3986744,-  3113
130,-28.86894408,-90.45062428,313.3690322,-219.9137655,-25.55727413,-396.2826784,179.258178,37.80114227,592.8925784,-53.48582855,144.7969107  3114
130,192.3263615,-290.1305735,-517.8364976,476.6559174,489.7345252,-40.49000759,180.0135956,-172.8726054,-166.4298378,-240.568055,-90.4263809  3115
130,917.7625263,142.119151,192.7047703,-107.155453,-286.4917367,-714.4369401,10.03613141,123.1584709,110.6519067,197.8729392,275.6513946,-85  3116
130,-825.5651246,-530.2628367,-39.54709655,-615.7184112,110.6715971,93.12998573,180.7282784,-90.19242849,-87.13249386,126.1719603,59.7942123  3117
130,-911.9678263,-285.8331617,-105.9158519,-439.3639048,135.2742305,-444.6524646,-45.58735016,670.2186075,-80.96356006,-307.5342642,112.89300  3118
130,-118.8950549,-86.91993822,301.0531429,-35.25596491,312.5046034,106.392939,-217.1467195,86.19504596,-198.8158681,-224.880377,-106.9814077,  3119
130,-957.1713667,-80.86933082,556.27,625.8261694,29.2115227,-215.5055135,405.3541557,339.3789473,-67.75692364,240.2025077,-124.3992543,202.6  3120
130,-1068.047533,-191.2119208,64.99137017,-693.30079,-33.52345841,304.8614587,-268.914528,14.78536142,83.39592467,78.57621677,-240.7179113,3  3121
130,110.51712,164.0304639,149.3924816,245.4289092,388.3005318,-30.3544971,38.43296436,-230.523076,192.3655112,-125.1417304,128.606591,44.529  3122
130,760.4293517,24.2700659,377.1286844,404.1263779,-207.4123502,-408.8650806,141.2405061,-244.0624321,1.221951411,-53.24457044,423.1302312,5  3123
130,-93.80836511,-732.4498279,222.0824847,242.8783697,60.34482198,48.963456,-293.1585327,181.656381,134.2035711,-164.3344737,-163.5902448,-2  3124
130,-590.7786593,418.877427,431.5078825,307.8058221,-390.4706457,-269.3984279,-706.6841322,165.1463406,133.7347064,-48.51551568,154.2730481,  3125
130,-514.812354,31.99676567,76.32827271,-634.4341084,153.1397874,-218.5278378,204.498593,688.2669008,-143.3999951,231.2733021,398.6411139,-2  3126
130,359.8932728,731.1461685,59.31117095,-491.0733646,-142.9936921,454.3251206,-159.3897212,119.5165243,-70.83835411,-121.0015499,167.0997959  3127
130,-266.5503203,-404.7829346,-245.9348049,-563.1795118,-671.2529843,-378.6783631,230.1196843,52.11555061,69.81843916,-256.2791476,152.15979  3128
130,-564.9610813,-1082.38032,-228.5147419,285.6804967,-209.8743374,54.05226271,-40.96995744,-210.2737825,165.5499053,98.57275734,220.9650449  3129
130,480.7342071,245.0814251,274.6091384,-28.53232121,-261.026929,90.77626232,566.0948083,-26.50007066,162.1080885,213.3700184,-19.90252513,-  3130
130,652.4694962,-486.4270538,584.5299105,578.1080388,127.6218459,42.12058817,93.90714403,-0.369428117,46.36005505,-2.03194933,221.0194918,1  3131
130,-96.05726087,1268.124154,407.5216761,59.20963487,54.95101812,-175.4343367,-73.26703537,222.3535755,233.9743735,-197.8928889,76.85250752  3132
```

By combining preprocessing techniques, advanced feature extraction methodologies, and a powerful hybrid neural network, this approach aims to address the challenge of image classification across diverse categories within the Coco dataset comprehensively. The model was compiled using the Adam optimizer, sparse categorical cross-entropy loss, and accuracy as the evaluation metrics. It was trained for ten epochs on the training dataset. During training, the model updates its parameters to minimize losses and improve the accuracy. The training history monitored the loss and accuracy metrics, and the performance of the model was evaluated. The iterative training process allowed the model to learn the patterns of the dataset and improve its predictions, as shown in Table5.

Table 5. Performance of the training processing of proposed hybrid CNN_LSTM model the COCO dataset

Feature	Value									
Epoch	10	20	30	40	50	60	70	80	90	100
Loss	0.122	0.041	0.0269	0.0184	0.0139	0.0104	0.0078	0.0074	0.0063	0.0067
Accuracy	0.9626	0.9873	0.9915	0.9941	0.9954	0.9961	0.9973	0.9976	0.998	0.9978

In training a neural network, loss is a crucial metric because it shows how well the model learns by reducing errors. The loss is depicted in Figure 6, which decreases gradually as the number of epochs increases. Relying on this behavior is necessary because the predictive power of the model would increase. For instance, one can observe that the loss decreases from a starting point of 0.122 and ends at only 0.0063 after hundred epochs. The significant decrease in loss indicates that the model captured the fundamental correlations and characteristics of the Coco dataset.

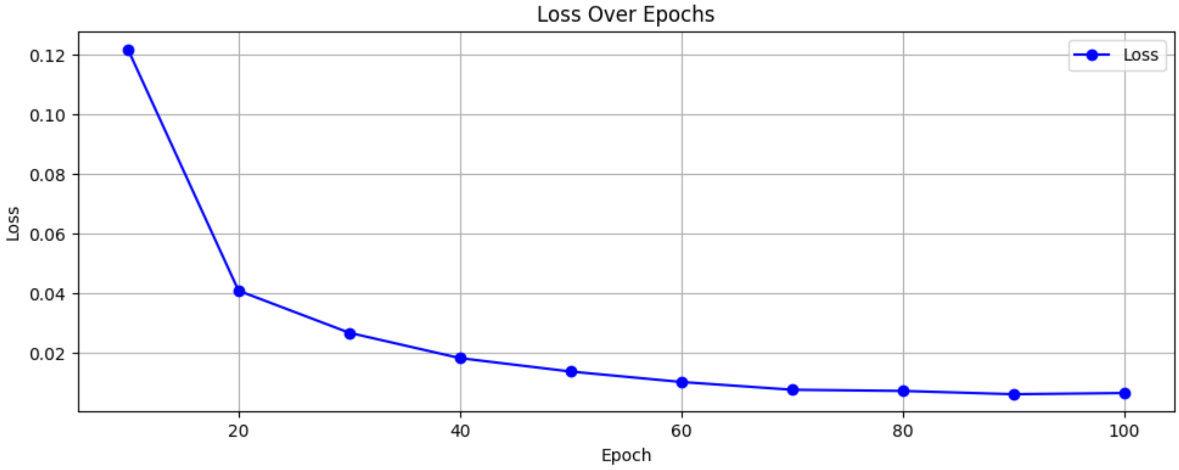


Figure 6. Loss results over epochs.

Another important measure is the accuracy, which shows the percentage of correctly categorized instances out of the total instances in a dataset. As shown in Fig. 7, an impressive and steady increase was evident throughout the training period. The model began with an accuracy of 0.9626 after epoch ten and continued until it achieved an accuracy of 0.9978 at epoch 100. The fact that this curve shows an improvement implies that the hybrid CNN-LSTM model is acquiring more accuracy in classifying objects contained in the COCO dataset.

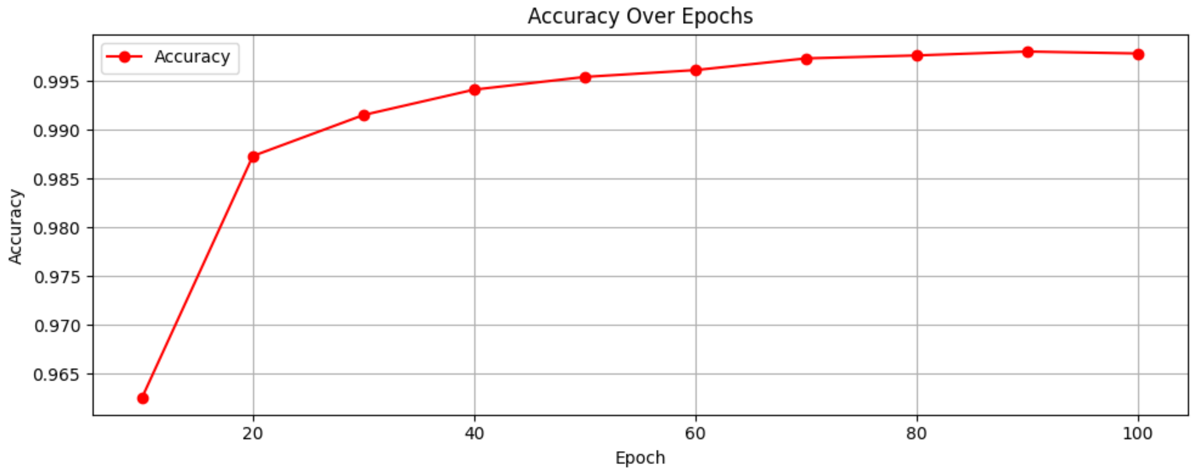


Figure 7. Accuracy results over epochs.

Table 5 shows the training results for the COCO datasets, demonstrating the effectiveness of the proposed hybrid CNN_LSTM model. The loss consistently decreased with each epoch, indicating improved prediction and feature extraction capabilities. The accuracy steadily increased, indicating the model's ability to recognize patterns and make accurate classifications. In the Figure 8 apply the proposed hybrid CNN_LSTM model to detection and classify objects

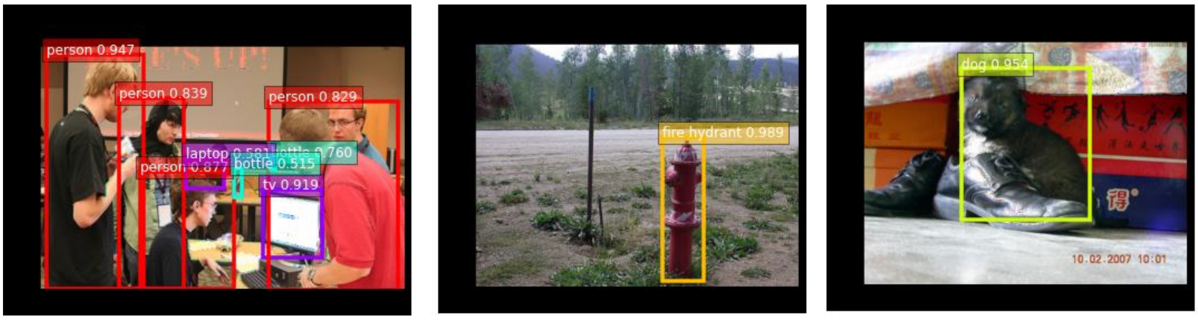


Figure 8. Sample of detection and classify objects based on proposed hybrid CNN_LSTM model

Overall, the model successfully learned and performed well on all datasets, captured their unique characteristics, and achieved high accuracy values. In the test phase(as shown in Table 6), the proposed hybrid CNN_LSTM model demonstrated high accuracy and strong performance on the COCO dataset, achieving an accuracy of 0.9917, along with impressive precision, recall, and F1 score values. These results emphasize the varying complexities and performance levels of the model across the COCO dataset.

Table 6. The performance of proposed model vs machine learning models

Models	Accuracy	Precision	Recall	F1 Score

SGDClassifier	0.21	0.49	0.21	0.19
GaussianNB	0.52	0.58	0.52	0.48
MLPClassifier	0.81	0.82	0.81	0.81
hybrid CNN_LSTM	0.9917	0.991738	0.9917	0.991695

This illustrates that the Hybrid CNN-LSTM model significantly outperforms traditional machine learning models, showing superior accuracy, precision, recall, and F1 Score. This underscores the effectiveness of deep-learning approaches in complex classification tasks.

TABLE 7. Performance Evaluation Of Different Classifiers.

Algorithms		**Year**	**Precision**
Puri [39] **.**	COCO dataset + RESNET50-SEGNET	2019	0.234
	COCO dataset + VGG16-UNET		0.458
	COCO dataset + VGG16-PSPNET		0.491
	COCO dataset + Mask R-CNN		0.332
Sharma [40]	COCO dataset	2021	0.8
Jintasuttisak et al[41]	COCO + YOLOV3	2022	0.95
Proposed	hybrid CNN-LSTM	2023	0.9917

As shown by Table 7, it presents the outcomes regarding the development of recognition and classification of objects through COCO dataset since its development. The most remarkable feature of the hybrid CNN-LSTM model proposed by the author in 2023 is that it significantly surpasses any previous model's accuracy with 0.9917. The efficiency of correctly recognizing items within the database is improved considerably by this.

The sequence of improving precision in the early models up to better precision in new approaches indicates the ongoing progress in deep learning technologies and neural networks designs. The hybrid CNN-LSTM model that attained a precision score of 0.9917 proved highly efficient in object recognition, and such capability could serve multiple practical purposes.

Conclusion

This paper, present a new classification strategy based on a mixed CNN-LSTM system which is more sophisticated than different varieties of each category in the COCO dataset. Data preparation entails thorough process preparation comprising critical measures of greyscale transformation, contrast enhancement, and noise suppression. Thereafter, critical feature extraction methods such as Principal Component Analysis (PCA), Gray-Level Co-Occurrence Matrix (GLCM) and Histogram of Oriented Gradients (HOG) are utilized to enhance robust image classification. The main innovation is centered on designing an advanced CNN-LSTM model that can read spatial and temporal trends contained therein. This architecture consists of vital building blocks which include conv_1d, max_pool, lstm_1, and dense1 which provides excellent accuracy scores on the COCO dataset. It is worth mentioning that the model manages a high F1 score of 0.999949, with an accuracy of 0.9917, precision at 0.991738, and a recall value at 0.99.

Moreover, the model shows monotonous reductions in losses as trainings and increases in accuracies indicating that it can catch very weak patterns in a dataset. This approach incorporates preprocessing techniques, advanced feature extraction methods, and a hybrid neural network architecture that can be termed as a powerful tool for complicated image categorization problems. This study presents a valid process of classification of large-scale images, which has shown remarkable efficiency in various real scenarios.

References

[1] Wang, X., Wang, A., Yi, J., Song, Y., & Chehri, A. (2023). Small Object Detection Based on Deep Learning for Remote Sensing: A Comprehensive Review. *Remote Sensing*, *15*(13), 3265.

[2] Chengkang, W., & Longxin, Z. (2023). A Survey of Object Detection Models Based on Deep Learning. *Computer Science and Technology*, *2*(2), 93.

[3] Li, W., Dong, X., & Wang, Y. (2021). Human emotion recognition with relational region-level analysis. *IEEE Transactions on Affective Computing*.

[4] Sampath, V., Maurtua, I., Aguilar Martin, J. J., & Gutierrez, A. (2021). A survey on generative adversarial networks for imbalance problems in computer vision tasks. *Journal of big Data*, *8*, 1-59.

[5] Kotkar, V. A. (2020). Scalable anomaly detection framework in video surveillance using Keyframe extraction and machine learning algorithms. *Journal of Advanced Research in Dynamical and Control Systems*, *12*(7), 395-408.

[6] Ozdemir, A., & OZKAN, I. A. (2023, September). Classification of Unmanned Aerial Vehicle and Bird Images Using Deep Transfer Learning Methods. In *Proceedings of the International Conference on Advanced Technologies* (Vol. 11, pp. 189-196).

[7] Willi, M., Pitman, R. T., Cardoso, A. W., Locke, C., Swanson, A., Boyer, A., ... & Fortson, L. (2019). Identifying animal species in camera trap images using deep learning and citizen science. *Methods in Ecology and Evolution*, *10*(1), 80-91.

[8] Yudin, D., Sotnikov, A., & Krishtopik, A. (2019, September). Detection of big animals on images with road scenes using Deep Learning. In *2019 International Conference on Artificial Intelligence: Applications and Innovations (IC-AIAI)* (pp. 100-1003). IEEE.

[9] Li, P., Chen, P., & Zhang, D. (2022). Cross-modal feature representation learning and label graph mining in a residual multi-attentional CNN-LSTM network for multi-label aerial scene classification. *Remote Sensing*, *14*(10), 2424.

[10] Hussan, M. I., Saidulu, D., Anitha, P. T., Manikandan, A., & Naresh, P. (2022). Object detection and recognition in real time using deep learning for visually impaired people. *International Journal of Electrical and Electronics Research*, *10*(2), 80-86.

[11] Saurav, S., Gidde, P., Saini, R., & Singh, S. (2022). Dual integrated convolutional neural network for real-time facial expression recognition in the wild. *The Visual Computer*, 1-14.

[12] Bi, Y., Li, D., & Luo, Y. (2022). Combining keyframes and image classification for violent behavior recognition. *Applied Sciences*, *12*(16), 8014.

[13] M. Murugesan, K. Kaliannan, S. Balraj, K. Singaram, T. Kaliannan, and J. R. Albert, “A Hybrid deep learning model for effective segmentation and classification of lung nodules from CT images,” Journal of Intelligent and Fuzzy Systems, vol. 42, no. 3, 2022, doi: 10.3233/JIFS-212189.

[14] SAROJA, M., & Mary, A. B. (2023). Image Captioning Using Improved YOLO V5 Model and Xception V3 Model.

[15] Sun, M., Zhao, J., & Shang, H. (2020). Building Energy Consumption Prediction with Principal Component Analysis and Artificial Neural Network.

[16] Saxena, A. (2022). An Introduction to Convolutional Neural Networks. *Int. J. Res. Appl. Sci. Eng. Technol*, *10*(12), 943-947.

[17] Deng, M. (2020). Robust human gesture recognition by leveraging multi-scale feature fusion. *Signal Processing: Image Communication*, *83*, 115768.

[18] Zhang, L., Sheng, G., Hou, H., & Jiang, X. (2020, June). A fault diagnosis method of power transformer based on cost sensitive one-dimensional convolution neural network. In *2020 5th Asia Conference on Power and Electrical Engineering (ACPEE)* (pp. 1824-1828). IEEE.

[19] Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., ... & Ghayvat, H. (2021). CNN variants for computer vision: History, architecture, application, challenges and future scope. *Electronics*, *10*(20), 2470.

[20] Ren, Y., Yang, J., Zhang, Q., & Guo, Z. (2019). Multi-feature fusion with convolutional neural network for ship classification in optical images. *Applied Sciences*, *9*(20), 4209.

[21] Oruh, J., Viriri, S., & Adegun, A. (2022). Long short-term memory recurrent neural network for automatic speech recognition. *IEEE Access*, *10*, 30069-30079.

[22] Hans, A. S. A., & Rao, S. (2021). A CNN-LSTM based deep neural networks for facial emotion detection in videos. *International Journal Of Advances In Signal And Image Sciences*, *7*(1), 11-20.

[23] Le, X. H., Ho, H. V., Lee, G., & Jung, S. (2019). Application of long short-term memory (LSTM) neural network for flood forecasting. *Water*, *11*(7), 1387.

[24] Lin, T. Y., Patterson, G., Ronchi, M. R., Cui, Y., Maire, M., Belongie, S., ... & Dollar, P. (2017). COCO dataset. *COCO Consortium*.

[25] Alkalai, M., & Lawgali, A. (2020, December). Image-preprocessing and segmentation techniques for vehicle-plate recognition. In *2020 IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS)* (pp. 40-45). IEEE.

[26] WINATA, H. N., NOGUCHI, R., TOFAEL, A., & NASUTION, M. A. (2019). Prediction of microalgae total solid concentration by using image pattern technique. *Journal of the Japan Institute of Energy*, *98*(5), 73-84.

[27] Welsh, T., Ashikhmin, M., & Mueller, K. (2002, July). Transferring color to greyscale images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (pp. 277-280).

[28] Rao, B. S. (2020). Dynamic histogram equalization for contrast enhancement for digital images. *Applied Soft Computing*, *89*, 106114.

[29] Tan, S. F., & Isa, N. A. M. (2019). Exposure based multi-histogram equalization contrast enhancement for non-uniform illumination images. *Ieee Access*, *7*, 70842-70861..

[30] Jaiswal, G., Sharma, A., & Yadav, S. K. (2021). Critical insights into modern hyperspectral image applications through deep learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *11*(6), e1426.

[31] Mutlag, W. K., Ali, S. K., Aydam, Z. M., & Taher, B. H. (2020, July). Feature extraction methods: a review. In *Journal of Physics: Conference Series* (Vol. 1591, No. 1, p. 012028). IOP Publishing.

[32] Cao, L. J., & Chong, W. K. (2002, November). Feature extraction in support vector machine: a comparison of PCA, XPCA and ICA. In *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02.* (Vol. 2, pp. 1001-1005). IEEE.

[33] Zhao, L., Li, J., & Ren, H. (2020, June). Multi domain fusion feature extraction and classification of ECG based on PCA-ICA. In *2020 IEEE 4th information technology, networking, Electronic and automation control conference (ITNEC)* (Vol. 1, pp. 2593-2597). IEEE.

[34] Chekouo, T., Mohammed, S., & Rao, A. (2020). A Bayesian 2D functional linear model for gray-level co-occurrence matrices in texture analysis of lower grade gliomas. *NeuroImage: Clinical*, *28*, 102437.

[35] Mirkes, E. M., Bac, J., Fouché, A., Stasenko, S. V., Zinovyev, A., & Gorban, A. N. (2022). Domain Adaptation Principal Component Analysis: base linear method for learning with out-of-distribution data. *Entropy*, *25*(1), 33.

[36] Barbu, T. (2022, September). Multiple Pedestrian Tracking Framework using Deep Learning-based Multiscale Image Analysis for Stationary-camera Video Surveillance. In *2022 IEEE International Smart Cities Conference (ISC2)* (pp. 1-7). IEEE.

[37] Sachar, S., & Kumar, A. (2021). Survey of feature extraction and classification techniques to identify plant through leaves. *Expert Systems with Applications*, *167*, 114181.

[38] Sundaram, N., & Meena, S. D. (2023). Integrated animal monitoring system with animal detection and classification capabilities: a review on image modality, techniques, applications, and challenges. *Artificial Intelligence Review*, 1-51.

[39] Puri, D. (2019, September). COCO dataset stuff segmentation challenge. In *2019 5th international conference on computing, communication, control and automation (ICCUBEA)* (pp. 1-5). IEEE.

[40] Sharma, D. K. (2021, March). Information Measure Computation and its Impact in MI COCO Dataset. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)* (Vol. 1, pp. 1964-1969). IEEE.

[41] Jintasuttisak, T., Edirisinghe, E., & Elbattay, A. (2022). Deep neural network based date palm tree detection in drone imagery. *Computers and Electronics in Agriculture*, *192*, 106560.