Improved Multi-Label Image Classification Performance using Supervised CNN-LSTM Deep Neural Network

Joseph James S

Research Scholar, School of Computing, SRM Institute of Science and Technology,
Chennai, Tamil Nadu-603203, India
josephjs@srmist.edu.in

Lakshmi C

Professor, School of Computing, SRM Institute of Science and Technology,
Chennai, Tamil Nadu-603203, India
lakshmic@srmist.edu.in

Abstract

Deep Convolutional Neural Network (CNN) classification of single-object image labels has shown high efficiency. However, the great bulk of actual application data comprises of multiple label object images that belong to a variety of scenes, objects, and actions in a single image. Most of the recent research studies on multiple object label classification rely on individual classifiers for each label category and use probability ranking for final classification. These methods already in place work better, but they cannot find the dependencies between multiple labels in an image. In this paper, we use deep CNN architecture and long short-term memory (LSTM) to solve the problem of label dependence. Our proposed CNN-LSTM methodology learns the embeddings of object label to depict semantic object label dependence and image label association using a robust multi-label classifier cost function (RMLC), which is a ramp loss function. The feature extraction is carried out by a convolution neural network (CNN) pipeline; whereas multi-object label correlation is identified by LSTM using object labels and features extracted from input images. We use the loss function to make sure that correlated labels and corresponding features map close to each other, limiting the high value updation of weights for the images with improper labels, and the object label prediction progresses every time, which helps to improve the multi-label learning task. Experiments conducted using the proposed framework on benchmark visual recognition datasets such as CIFAR-10, STL-10, PASCAL VOC 2007, MS-COCO, and NUS-WIDE provided performance comparatively better than many existing methods in terms of accuracy and mean average precision. The CNN-LSTM + RMLC achieve the best test accuracy of 95.56 % on the STL dataset, which is 4% higher than the existing method, and the best mean average precision (mAP) of 82.6 on the MS-COCO dataset, which shows the feasibility and usefulness of our suggested framework on multiple label image classification.

Keywords*: *Multi-label image, label correlation, image classification, LSTM, ramp loss, CNN.

1. Introduction

Multiple label image categorizations are a difficult and crucial task in machine vision with numerous real-world applications. The majority of picture classification research focuses on single item label classification. However, a single object label in a picture is unimportant in many real-world application data sets, which typically contain several object labels with extensive semantic information to explain scenes, activities, and their interactions. The number of object labels available in an image varies from image to image. In Fig.1, for example, the picture labels match to item labels such as car, horse, and human. Understanding multiple item label images requires training the semantic information and their label dependencies. Convolutional Neural Network (CNN) is a popular solution for multiple item label classification that is used to reduce the problem to a single object label prediction problem and train using cross entropy loss [1] [2]. However, when the numerous object labels are considered separately, the approach fails to train on the dependency between multiple object labels. The solution to the problem of multiple label classification is to efficiently learn the significant feature of every object in the input image and incarcerate the label dependency, which can increase the classification model's efficiency. With CNN's prominent feature depiction capacity [46] and a plethora of labeled datasets like ImageNet [48], deep convolutional neural network models have made significant progress in multiple label prediction issues. These methods overlook three

issues: picture object scale fluctuation, label correlation, and label correlation between image characteristics and labels.

Many previous studies have shown that multiple item label prediction tasks demonstrate substantial label dependency features [3]. For example, a ship and water will always show together, but water and a bus will rarely appear together. Many prior research employed graphical models [3] to represent label dependency, with the co-occurrence dependencies with pair wise co-occurrence probabilities being the preferred strategy of the model to predict the final label probability. These strategies are either incapable of learning higher order correlations [3], or need a high level of computational complexity to learn big complex label dependencies [4]. We use LSTM to represent multi-object label dependencies in the proposed study to capture multi-label interactions while maintaining computation cost acceptable. We demonstrate that LSTM improves task accuracy in categorization. The majority of past work on multi label image classification using CNN employs the same picture attributes for all classifiers. However, using the same characteristics to categorize numerous item labels overlooks the image's small objects [5]. Another issue with multi-label photos is that some of the labels are erroneous because they are collected by an automated method, resulting in missing and noisy labels.

Fig 1 Multiple object image

To avoid the aforementioned issues, we constructed LSTM in our proposed method to recognize image characteristics based on past classification results. The purpose is to train the model implicitly using kernel filters on small object portions in photos, such that the convolutional neural network may focus on different regions of the image to recognize distinct item labels. For example, during the classification of numerous object labels in Figure-1, the proposed method prioritizes smaller things (e.g., automobile, hat) after detecting major objects such as (i.e. horse, Person). Minor object labels are difficult to detect by it, but they can be easily inferred if relevant context information is provided. The major contribution of our proposed work are abridged as follows,

1) The amalgamation of deep CNN and LSTM resolves the problem of multifaceted feature extraction and correlation between labels in multiple label image classification.
2) We design robust ramp loss function that integrated with CNN-LSTM framework. The designed loss function aimed at to exploit label correlation thereby provides optimal weight for the training samples with noisy and incomplete labels.
3) Our proposed classifier framework CNN-LSTM-RMLC can also work along with hand crafted features.

This work is divided into five sections, with the first providing an introduction to multi-label picture categorization, its scope, and its significance in research. It also includes the primary scientific contribution and benefits of this study. Section 2 discusses the research background, relevance, deep learning capabilities, and current advances of the convolution neural network. It also discusses issues with current multi-label picture classification algorithms. Section 3 describes the suggested approach, CNN-LSTM, loss function, design parameters, and implementation details. It also includes the CNN-LSTM-RMLC network model structure and training methodologies. Section 4 compares experimental results on several image databases such as CIFAR-10, STL-10, MS-COCO, NUS-WIDE, and PASCAL VOC. It examines the benefits of the suggested strategy over noisy data and provides qualitative results. Section 5 concludes this work and makes predictions for future research on the same topic.

2. Related Works

The classification of multi-object picture labels is an important objective in learning algorithms, with applications in computer vision, music retrieval, and text categorization. Image classification research has grown in popularity as a result of the availability of large amounts of human-labeled data, such as ImageNet [6] and CIFAR-10, as well as recent advancements in deep learning techniques, particularly CNN. Recent work on multiple object image label classification using CNN with top-k ranking optimization provides minimal weights to losses and achieves good results [1]. To converge probability values from several region proposals derived from the input image, Wei *et al* [5] employed max-pooling. In recent years, deep learning-based multiple label categorization of image data has considerably benefited a variety of image-related applications such as semantic

segmentation [7], object detection and recognition, and activity detection [8]. Yu *et al* [9] proposed a deep dual flow network with multiple instances to exploit information on both local and global levels. Creating a list of co-occurring objects can be thought of as a multi-label categorization issue. Depending on its attention capacity, a model may isolate specific elements of an image while ignoring others. It's been used in image processing for a long time. For example, SENet [10] considers channel linkages, enhances feature channels with attention mechanisms, intelligently determines the significance of each feature channel, emphasizes key characteristics, and suppresses dull features. Woo *et al.* [11] introduced GSoP-Net to capture more discriminative representations than SENet [10] and CBAM [11], which use the attention strategy of feature channel and feature space combination, by utilizing the global picture of the deep CNN's second order statistics. Through repeated procedures, each pixel maintains its long-term reliance, and CCNET [12] accesses each pixel's relevant data via a crises-cross path.

Sumbul *et al.* [13] used a bi-directional LSTM-based attention technique to estimate many attention scores in order to measure the relevance levels of various picture local areas. The classifier model predicts labels for each image using a global descriptor specified by the attention ratings. Li *et al.* [14] suggested a method that improves the channel-wise attention mechanism by merging regional attention maps and relative labels, both of which are employed to build label sequence in large-scale data, using a CNN-RNN-based bi-model multi-label learning framework. To investigate pathological connections in the multi-label chest X-ray (CXR) image classification job, Chen *et al.* [15] created a novel label co-occurrence learning system based on Graph Convolution Networks (GCNs). By introducing pathological word embedding and multi-layer graph information propagation to fine-tune pathology prediction belief states, the link between pathologies is expanded into a collection of classifier scores. Park *et al.* [16] presented MarsNet, a CNN-based end-to-end network for multi-label categorization that can accept inputs of varying sizes. To allow the network to accept such images, the dilated residual network (DRN) is updated to generate higher-resolution feature maps, and horizontal vertical pooling (HVP) is built from the ground up to collect positional information from the feature maps efficiently. The multi-label score module and the threshold estimation module are also utilized for multi-label classification.

Exploring image similarities and optimizing visual feature embedding for multi-label chest X-ray image classification was suggested using a graph convolutional network-based semantic similarity graph embedding framework [17]. Multiple label categorizations have long been investigated in semi-supervised learning in order to reduce human tagging effort. Semi-supervised learning approaches in image analysis are grouped into three types: 1) A technique based on adversarial learning [18], [19], and [20]; 2) A graph-based technique [21]; and 3) A label propagation-based strategy Dong *et al.* provided a framework for searching neural network models and automated the creation of neural architecture for classification models using gradient-based optimization for adversarial medical picture segmentation. Semi-supervised deep convolutional GANs and a retinal picture synthesizer were proposed by Diaz-Pinto *et al.* [19] for glaucoma evaluation. This method automatically creates labels and has high glaucoma classification accuracy. Iscen *et al.* [22] employed a transductive label propagation technique based on various assumptions to forecast outcomes and generate pseudo-labels for unlabeled input to create a neural network model. The transductive technique is based on the nearest neighbor graph of the dataset, which was produced using network embedding. These studies rarely employ deep neural network approaches since they focus on conservative features. The mapping of label associations has also been disregarded or overlooked during classifier learning. Wang *et al.* [23] suggested a framework for multiple labels was made. SDRL is a strategy for semi-supervised dual-relationship learning. SDRL is intended to discover latent relations in given samples, such as instance-level relationships in feature space between labeled and unlabelled, as well as label-level relationships within each sample. To properly align the shifted feature distributions, a two-classifier domain adaptation structure is used. To summarize, past research attempts on multi-label picture classification that we are aware of either fail to generate context or use a Convolutional Neural Network-based direct label classification module. Existing models necessitate information about the number of class labels associated with image objects. The suggested model incorporates direct recognition, context, label estimation, and object label correlation in an image.

In this article, we suggest a CNN-LSTM model with a ramp loss function for multiple object image label classification that efficiently learns label co-occurrence dependency in the images. The multiple label CNN-LSTM framework learns an image object label embedding to become acquainted with the relationship between image features and labels. The object label embeddings are derived from a CNN for each label. The label co-existence dependency on vector space is modeled with the use of LSTM neurons, which keep a record of information about label context in the memory gate. The Long Short Term Memory unit calculates the probability of a multiple object label classification successively, where the probability of an object label at each time stamp is calculated with the image label embedding vector and the output provided by LSTM neurons. In final classification, a softmax activation algorithm can be used to find the multiple object label classification with the highest probability. The proposed CNN-LSTM framework with ramp loss function is a better framework has following advantages:

- LSTM neurons are more compact and powerful to model label co-existence dependency than other models used for label dependency.
- The robust ramp loss functions in the weighted approximate ranking method to capture the multiple object label dependency and to exploit label correlation.

The proposed CNN-LSTM model with a robust ramp loss function was evaluated using experiments on multiple object label datasets such as PASCAL VOC 2007, Microsoft COCO, NUS-WIDE, and single object label datasets like the CIFAR-10 dataset and STL-10 dataset. The output of multiple object image label classification is represented as a one-hot-encoded form of a binary vector where each bit represents the existence or non-existence of an object label in the input label. The results show that our proposed framework is more effective than existing methods for classifying objects with more than one label.

3. Proposed Work

3.1. Feature extractor

Feature extraction can be done with any standard convolutional neural network architecture used for single-object image label classification. In our proposed implementation, we used a convolutional neural network with weight parameters from ResNet-101 pre-trained using the ImageNet dataset. The feature map representations from the final convolutional layer of CNN are given as input to LSTM for label classification and embedding space branches. Let I = { X_1, X_2, ……X_k }^{d*k} denotes the collection of images with respective labels L = { L_1, L_2, ……L_k }, L {0, 1 } C*k, where Li is a one-hot encoded vector of the image labels with a size of C. X_i, k, C, and d are the image dataset size, the total number of labels, and the dimension of the image data, respectively. In the one-hot encoding of labels, L_{in} = 1 if the X_i image has the n^{th} object label, otherwise L_{in} = 0. We give the image data X to the feature extraction model, which is a CNN input layer, to acquire the feature map representations FR_x from the image. Using the Re-LU activation function, a convolutional neural network applies feature detectors to the input image to build feature maps. Feature filters aid in the extraction of various features in an image, such as bends, edges, horizontal lines, and vertical lines. The feature maps are then pooled to ensure translation invariance. Pooling is predicated on the idea that changing the input by a little amount has no effect on the pooled outputs. We employed maximum pooling, which performs better than minimum or average pooling. The feature maps generated from PASCAL VOC 2007 using CNN are shown in Fig. 2.

Fig 2. Original Image Conv2 Conv5 Conv8

3.2. Convolutional neural network

3.2.1. Convolutional layer

Using multiple feature detectors, the convolutional layer captures the features of interest from the input image and provides feature maps. The neurons in the first convolutional layer extract simple features such as edges. Following convolutional layers, neurons extract high-order features from the input image. Each convolutional layer kernel may extract features from the input image, and neurons are allocated to different sections of the image to create feature maps.

3.2.2. Activation layer

Any neural network must have non-linearity in order to be effective. A non-linear activation function, in addition to perceptrons, is required to shatter the linear mixing of input and make a model a general approximator of continuous functions. Later, Jarrett et al. included rectified linear units (ReLu) in CNNs to improve model performance [8]. ReLu is well-known for its non-linear activation of convolutional layer outputs. After calculating a weighted sum and a bias, the activation task is applied to the hidden layer. The CNN-LSTM model achieves

non-linearity by passing the result of the convolution process via the ReLu function. As a result, the quantities in the resulting feature maps are the ReLu function applied to them, rather than sums. Hidden layers in a trained CNN correspond to numerous abstract representations of input attributes. When presented with an unknown input, a CNN has no way of predicting which of the abstract representations it has learnt will be meaningful. Any neuron in the deep layer that reflects a learnt abstract representation can either be relevant or irrelevant. If the neuron is unimportant, this does not rule out the possibility of other viable abstract representations. All learnt abstract representations are designed to be self-contained. As a result, CNNs with non-negative activation functions are preferred. The Rectified Linear function is the most prevalent, and a neuron that employs it is known as a ReLu. The ReLU activation is defined as: f(x) = max (0, x). The ReLu function has two significant benefits over the sigmoid function: 1) It is extremely simple to compute because it simply requires a comparison of the input to the value 0. 2) Depending on whether the input is negative or positive, the derivative is either 0 or 1.

3.2.3. Pooling layer

Subsampling or pooling layers are commonly used to mediate between multiple convolutional layers. Subsampling has numerous advantages for CNNs. It also seeks to prevent over fitting by focusing on local data with a subsampling window, reducing data dimensionality. Reduced data dimensionality aids in the reduction of quantities calculated. Because a limited number of scaling or dislocations make no difference after subsampling, the pooling layer also provides invariance such as translation, scaling, and rotation. The max-pooling technique excites the highest result of each subsampling region (Px, Py) on the input image and reduces it by Px and Py on width and height, respectively.

3.2.4. Classification layer

The classification layer is the topmost layer of any neural network, and it aggregates the final extracted feature maps and outputs a column vector with each row representing a class label. The sum of the estimated probabilities for each category is represented by each element of the resultant vector. In general, CNNs have a fully-connected layer that evolved from the traditional machine learning design of feature extraction and classification. To calculate the likelihood of output classification, softmax for single-label images and sigmoid for multi-label images are activated above completely linked layers. Fig. 3 and Fig. 4 show how our proposed CNN-LSTM-RMLC neural network layers for classifying the CIFAR-10 dataset are put together. Except for the input layer, our proposed CNN-LSTM-RMLC has 14 layers, which are denoted by the letters Cx, Sx, Rx, and Fx, which stand for convolutional layer, subsampling layer, recurrent layer, and fully connected or dense layer, respectively, and x signifies the layer succession number. Every convolution layer in our proposed network has a kernel size of 3x3 and a stride length of 1. Pooling layers maximizes input in a 2x2 region with a stride of 2. The maximum value and the trainable parameters are multiplied together, and then a bias is added on top of that. The input layer is given a normalized image with the dimensions 32 by 32 pixels. C1 has 64 convolution kernels, resulting in 64x (9+1) = 640 trainable coefficients, and outputs 64 feature maps of size 32x32. This layer has a total of 9 x (1+1) x (32x32) x 64 = 655360 connections. C2 is made up of 64 plane convolutional kernels of size 3 x 3 that result in 36928 trainable parameters. S3 is made up of 64 sub-sampling kernels. As a result, there are 64x2 = 128 trainable parameters and a total of (2x2 + 1) x (16x16) x 64 = 81920 linkages. C4 and C5 are made up of 128 convolution kernels of size 3 x 3 with stride 1, resulting in 73856 and 147584 trainable parameters, respectively. S6 is made up of 128 subsampling kernels with a total of 128 x 2 = 256 trainable parameters. C7 and C8 have 256 and 128 convolution kernels, respectively, with kernel size 3 x 3 and stride 1, resulting in 295168 and 295040 trainable parameters. S9 is made up of 128 subsampling kernels, with a total of 128 x 2 = 256 trainable parameters. R10 is made up of 64 recurrent units that return the last output as a sequence to the next layer and generate 221952 trainable parameters. R11 and R12 are made up of 32 and 16 recurrent units, respectively, and return the last output as a sequence to the next layer, as well as 31104 and 10320 trainable parameters. F13 and F14 connect the output features set with its input, yielding 320 x 20+20 = 6420 and 210 trainable coefficients and connections, respectively. Softmax activation for single item classification is used in the output layer, which generates a 10 * 1 classification column vector with element 1 indicating the expected class label.

Algorithm 1: LSTM for Robust supervised Multi-label classification

Input: Feature Map vectors FV
Output: Predicted class labels $Y_{pi} = \{1, 2, \ldots C\}$
Initialization: d = 4096; S = 2048; C = Number of object classes for classification problem.
Procedure LSTM (FV, C, LSTM)
While $t < S$ *do*

$$Z^t = g(W_z FV^t + RE_z\, Y^{t-1} + b_z) \text{LSTM input}$$

$$I^t = \sigma\,(Wi\, FV^t + RE_i\, Y^{t-1} + P_i \otimes CE^{t-1} + b_i)\ \text{Input gate}$$

$$F^t = \sigma\,(W_f FV^t + RE_f\, Y^{t-1} + P_f \otimes CE^{t-1} + b_f)\ \ \text{Forget gate}$$

$$CE^t = Z^t \otimes i^t + CE^{t-1} \otimes f^t \text{Cell state (memory)}$$

$$O^t = \sigma (W_O FV^t + R_O Y^{t-1} + P_O \otimes CE^t + b_O) \quad \text{Output state}$$
$$Y^t = h(CE^t) \otimes O^t \quad \text{LSTM output}$$
$$v^t = h_t (Y^t) \quad \text{Dense layer output}$$

END
$EX = AP (v^t, v^{t-1}, v^{t-2}, \ldots. V^S)$ *Average pooling*
Compute $P_k = \{ P_1, P_2, \ldots\ldots, P_k\}$ *<- softmax(EX)*
Find Idx $= Support (max(P_k))$ *Highest Probability*
$Y^{'} = Idx$ *Label Prediction*
END Procedure

Where, $\otimes$ is the product with gate value in the activation function and W learned weight parameter matrices. In our proposed work, we use the activation function rectified linear units (ReLU) [25]. LSTM input along with weight and corresponding labels will be available in Z^t which is given to the input gate (I). The input gate concatenates the current input with the previous input and passes it to the forget function to get F. The forget gate determines which elements to erase or keep in the current input and sends data to cell states (CE), where the tanh operation with weights learned is performed. LSTM output (Y^t) generated with the use of hidden cell state values is a dot product with the output state of data. This LSTM output is passed to the fully connected layer (v^t) of a recurrent network. The class label probability is calculated from the average pooling (AP) of dense layer output. The softmax layer calculates the probability of a possible class label in an image, and an output class label (y) will be assigned based on the maximum probability in each class.

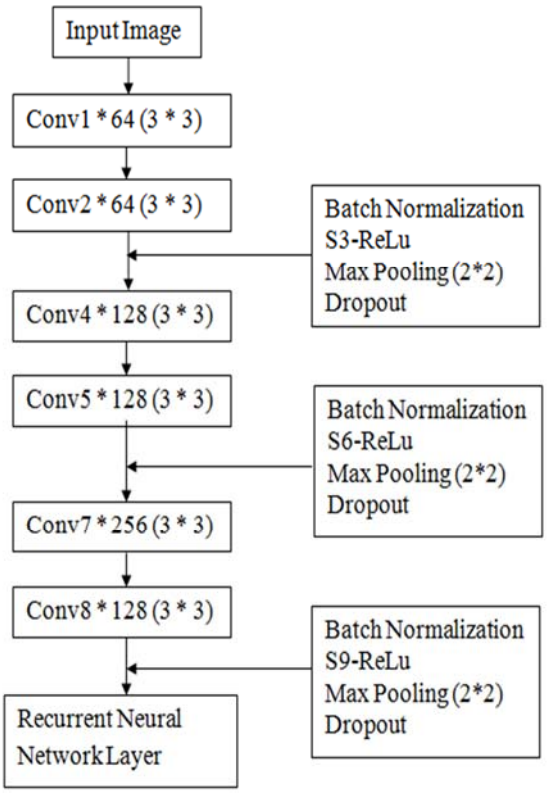


Fig 3. Architecture of deep CNN. Conv = convolutional layer, the x in Conv * x represents the total number of convolution kernels in the layer, the size of pooling window is y * y.

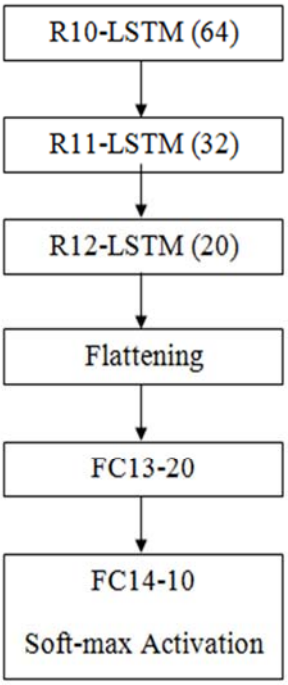


Fig 4. Architecture of recurrent neural network. LSTM = Long Short Term Memory, FC = dense or fully connected layer. The m in R-(m) represents the total number of recurrent units in the layer, the z in FC-(z) represents total number of neurons in the FC layer.

3.3. Long short term memory (LSTM)

The internal states of a recurrent neural network are utilized to imitate the temporal behavior of sequential input via circular connections in its internal neuron units. It can simulate long-term time-based dependency. LSTM is a basic RNN extension with three gates added to recurrent neurons. A forget gate F decides whether to maintain or erase the current state content, an input gate I fetches the input values, and an output gate O determines which values go to the output state. These three gates enable long-term memory to learn data dependency in the order, as well as simple optimization by using gates that lead the input values to transmit via the recurrent hidden layer

units R(t) while not interfering with the real output sequence. The problem of gradients exploding and going away with simple RNN is easily solved by LSTM [24].

3.4. Loss function

Machines employ a loss function, also known as an error function, to learn. It is a method for determining how effectively an algorithm models the training data provided. If the expected value differs greatly from the actual value, the loss function will vary by a considerable number (increase or decrease). Studies using progressive loss functions to reduce prediction error with the help of an optimization function. The suggested classification model employs the optimization problem given in Eq. (1) and represents collections of catalogues with positive and negative labels for sample data x_i.

$$\min \frac{\lambda}{2} trace(W^T W) + \sum_{i=1}^{l} \sum_{j=1}^{L_{xi}^{+}} \sum_{k=1}^{L_{xi}^{-}} F(r_j) R_s(W_j^T X_i - W_k^T X_i) + k \sum_{i=1}^{l} \sum_{j=1}^{m} R_s(y_{ij}(W_j^T X_i)) \quad \ldots\ldots\ldots (1)$$

Where F(.) is a function to calculate weight for various ranks, r_j is the rank of j^{th} class label *of* i^{th} image sample, the ramp loss function is $R_s(t)$ = minimum(1−*s*, maximum(0, 1−*t*)) and $-1 < s \leq 0$ is a factor which should be fixed by manipulator. W_j relates to j^{th} column of the weight matrix (W) with d x m dimension, λ is the regularization factor, which is also assigned by the manipulator, and k is a factor which limits the weight value of the loss function. The difference of two convex hinge losses can also be used to express the robust ramp loss function, $R_s(t) = H_1(t) - H_s(t)$, where $H_a(t) = max\ (0, a - t)$ is the classic hinge loss function. We use s = 0.75 in the proposed model experiments. The word F(.) is used to manage the ranking of different labels. The term in equation-1 is a regularization function that ensures that the model is used in conjunction with handcrafted features. The regularization function is omitted since our proposed framework is integrated with deep neural network architectures. The proposed approach, which differs from the WARP technique, uses a resilient ramp loss instead of the standard hinge loss function in optimization equation-1. Furthermore, the proposed strategy forces the classification model to produce positive values (> 1) for positive label data and negative values (< 1) for negative label data by utilizing the last term in optimization equation 1, which is not available in WARP. We used a weight function similar to that of [9, 12], which is as follows:

$$F(r) = \sum_{j=1}^{r} \alpha_j \text{ Where } \alpha_j = \frac{1}{j} \text{ and } r_j = \left\lfloor \frac{m-1}{n} \right\rfloor$$

where n is the number of images. The function F (.) used to control ranking labels throughout optimization. F (.) gives the loss value the least amount of weight if a positive class label is at the top of the label list and the most amount of weight if it is not at the top of the list. The hinge loss function gives a loss value that isn't limited by a range. Because of this, it gives a large loss value for labels that are wrong, which is far from the acceptable margin. These wrongly labeled data have a substantial impact on the weight matrix and reduce the predictive model's efficacy. The ramp loss function is preferable to the hinge loss function because it causes poorly labeled image data to yield only a limited degree of loss, despite of its placement related to the border. As a result, the improperly labeled data has no influence on the optimization process. We use the Adam optimizer to tackle the optimization problem. Because the execution time of the Adam optimizer is not directly influenced by the number of training samples for quadratic loss functions like ramp loss, our proposed framework is fast and performs well with vast volumes of data.

3.5. Model

The suggested CNN-LSTM-RMLC model is made up of three primary parts: The first is a feature extractor that uses convolutional neural networks, while the second is object label correlation that uses LSTM and picture characteristics and labels. Third, a classification module that combines feature extractor direct label identification results with LSTM object label correlation details. The feature extraction module extracts feature representations from input images using a typical convolutional neural network layer, which is then followed by an activation and max pooling layer for single object picture label classification. The recovered feature maps can be regarded of as abstract concepts of the objects in the input image. With these image feature maps, a thick layer-based direct label classifier for labels can be built on top of the feature representation. Deep CNN-layer extracted feature maps have significant semantic content and are abstract in nature. Fig. 5 displays the proposed framework's general architecture. The LSTM network architecture calculates the object recognition probability. To model the image and label relationship, the image feature representations and class labels are mapped in an embedded space. In the embedded space, the LSTM model serves as a dominant representation of the object-label correlation. At each time step, it uses predicted object label embeddings and maintains a hidden layer state to learn label co-existence evidence. The likelihood of a class label is known ahead of time, and class labels that have already been predicted are found using dot products of the sum of a picture's feature maps and embeddings of recurring layers.

The likelihood of a recognition track is attained as the product of earlier labels in the estimate path and the a-prior probability. A label l is denoted by a one-hot encoded vector as e_l = [0... 1, 0, 0 . . . 0], 1 at the l-th position, and 0 at the remaining location. The embeddings of the class label are obtained by multiplying the label embedded vector Um with the one-hot encoded vector. The label embedded value of class label l is available at l^{th} row of U_m.

$$W_l = V_m .e_l$$

The recurrent layer learns the correlation and dependencies in the hidden states using non-linear function as follows,

$$O(t) = h_o(q(t-1), w_l(t), q(t) = h_r(q(t-1), w_l(t))$$

Where q(t) is hidden states and o(t) is output state of recurrent layers at t time step. The class label implanting of the t^{th} label is $w_l(t)$.The recurrent network layer output and feature maps of images are mapped into the embedding space as follows:

$$x_t = h(V_o^x O(t) + V_I^x I)$$

Where V_o^x is projection of the recurrent neural network layer output and V_I^x feature map of the image. At last, the object label score calculated by reproducing the transpose of V_I and x_t to calculate the detachments between every class label embedding and x_t using $s(t) = V_m^T x_t$.

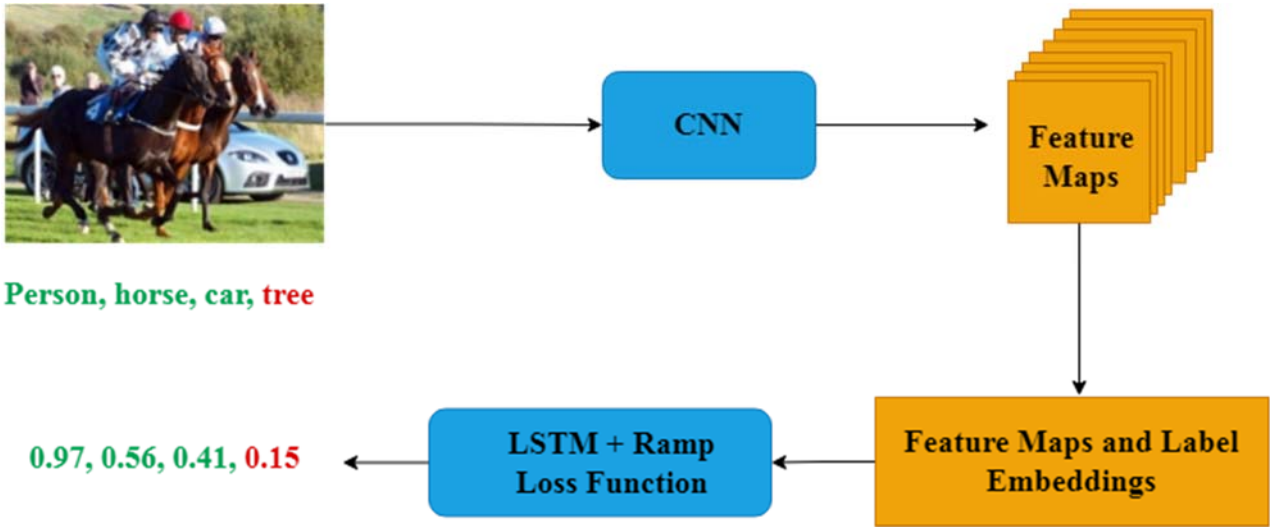


Fig 5. Proposed Model architecture Overview

The likelihood of predicted labels computed applying softmax function on the scores. For multiple labels sigmoid function used since the one class label probability calculation not dependent on other class label probability.

A prediction route is a set of class labels (c_1, c_2, c_3… c_N) for which the probability of each label c_t may be calculated using the image M's information and the previously predicted labels c_1, c_{t1}. The LSTM model predicts more than one label by finding the prediction path that maximizes the probability from the start.

$$C_1, \cdots, c_k = \arg\max P(c_1, \cdots, c_k|M)$$

$$= \arg\max P(c_1|M) \times P(c_2|M, M1)$$

$$= P(c_k|M, M_1, \cdots, M_{k-1}) \quad \text{----(2)}$$

Since the probability P (c_k|M, M_1,..., M_{k-1}), there is no best possible polynomial technique for determining the best possible prediction path. Eq. (2) is devoid of the Markov property. To identify the best forecast path, we use the beam search technique. At each time step t, the beam search algorithm finds the top-N most likely forecast paths as intermediate paths SP(t). It doesn't try to get the maximum likelihood label too soon.

$$SP(t) = \{P_1(t), P_2(t), \cdots, P_N(t)\}$$

We append N maximum likelihood labels to every intermediary path $P_i(t)$ at the t+1 time step, resulting in N x N paths in total. The N-likelihood paths are the intermediate paths that have the highest probability for the t+1 time step. The current intermediate paths have a lower probability than all of the candidate paths, which is the termination criterion for the beam search. It means that there are no more candidate paths with a higher probability.

Algorithm 2: CNN-LSTM -RMLC framework based solver for supervised multiple label classification

Feature Map Extraction
Input: Labelled training data X = { X_1, X_2, …….X_k } where k is the total number of images
Output: Feature Maps of given input images, probability score of labels
Procedure Feature Map Extraction (X, CNN)

1. *$X \rightarrow X^R$: Resize the colour image and convert it into gray scale image for easy processing*
2. *$X^R \rightarrow X^E$: Enhancing the edges of the images for better detection [43]*
3. *$X^E \rightarrow X^{np}$, Y : where X^{np} – image data in numpy array, Y – image object labels in the form of one-hot-encoding*
4. *X^{np}, Y →X_test, X_train, Y_test and Y_train : divide the data source into train set and test set*
5. *X_train, Y_train → CNN: The raw training data are sent to input layer of Convolutional neural network to extract feature representation maps.*
6. *For i = 1 to K do*
7. *$X\_train^{(i)} \rightarrow$ Conv2D : Raw training images are given to convolution to get important representation of the image provided kernel (5,5)*
8. *X_Conv2D → Leaky ReLu: Convolved images are passed to activation function Leaky ReLu to get the output*
9. *X_Conv2D_Relu →Maxpool: The activation outputs are given to max pooling to reduce the dimensionality of the processed image representation using (2, 2) size window.*
10. *H →{ H_1, H_2, …….H_k } : Where H is hidden layer and k total number of hidden layer.*
11. *$F^{(i)} \rightarrow$ H: Reduced convolved image data are passed through various hidden layers H to extract the high level feature representation(FR) of input images.*
12. *$FR^{(i)} \rightarrow$ V: Feature maps are converted into a feature vector V using time distributed manner.*
13. *$V^{(i)} \rightarrow$ LSTM: Feature vectors are passed to LSTM to get class label dependency and correlation.*
14. *L = { L_1, L_2, ……. L_m }*
15. *$V^{(i)}$_LSTM → L: Extracted feature vectors are given hidden layers of LSTM sequentially to get the correlation between labels via feature vectors.*
16. *$V^{(i)}$_LSTM_L → D: The last hidden layer output will be flattened and delivered to fully connected layer (Dense Layer D) for learning representations and labels along with label correlation.*
17. *D = { D_1, D_2, ……. D_n }*
18. *$V^{(i)}$_LSTM_L_D →Softmax: Learned features are given to softmax layer to get the output probability based on which classification of labels will be made. The error will be calculated between predicted label Y_{pi} and true label Y_i based on which gradient calculation will be done using back propagation to minimize the error.*
19. *End*

End Procedure

3.6. Model design parameters and implementation details

In our proposed technique trials, the convolutional neural network unit contains six convolutional layers and two fully linked layers, similar to the ResNet-101 network. The recurrent LSTM layer and the class label embedding vector space have sizes of 512 and 128, respectively. The user must specify the two parameters of our proposed model, the regularization parameter and the ramp loss function weight k, in order for the classification model to have a greater than zero for positive labels and a less than zero value for negative labels. The model is trained using an Adam optimizer with regularization parameters of 1e-4 weight decay and 0.9 momentums. We employ the k = 5 value in all of our studies, and the findings show that k values between 2 and 8 produce superior results. As a result, we set k to be the average of those two integers. In the experiments, 64-piece mini-batches were used. We also use learning rates of 0.005 and 0.3 dropout rates for all the hidden layers. The tests were done on a computer with an Itanium processor, 32GB of RAM, and 11 GB of Nvidia RTX 2080 Ti GPU memory.

4. Experiments

We tested our suggested technique on five visual object identification datasets: STL-10, CIFAR-10, MS-COCO, NUS-WIDE, and PASCAL VOC 2007. STL-10 and CIFAR-10 are data sources for single object picture labels. The results reveal that the proposed model outperforms several previous strategies in the field of multiple label classification.

4.1. Evaluation metric

We evaluate the performance of our proposed method for multiple object image label classification using seven metrics: mean average precision (mAP), overall precision and per-class (O-P, C-P), and recall scores (O-R, C-R), where the average is taken from all class categories and entire samples from the testing set, respectively.

The F1 score is calculated using the geometrical mean of recall and precision values (O-F1, C-F1). In addition, for multiple object image data, we compute the (mAP) measure [24]. We use the accuracy measure to evaluate performance on single-object label datasets.

4.2. CIFAR-10 dataset

The CIFAR-10 dataset is a single object label dataset that comprises 60000 image data points of size 32 x 32 in 10 different object categories such as automobile, airplane, bird, deer, cat, dog, horse, frog, truck, and ship. 50000 images are utilised for training the model, and training samples are divided into 10 folds. Ten thousand images are used for testing the model's accuracy. A training batch of size 64 is used. We performed two different sets of experiments. As a first method, we trained the model by initializing weights randomly from scratch. Secondly, we used model weights trained on the ILSVRC 2013 dataset initially and then fine-tuned the weights with the CIFAR-10 dataset. In order to make use of already trained model weights, we scaled up all images to size 256 ×256. An average accuracy of 10 folds is calculated as the final accuracy of the model. The obtained results through the experiment are shown in Table 1.

The existing approaches described in Table 1 use various deep neural network architectures, but all of them train and validate using the same experimental settings that we used in our experiment. Our suggested CNN-LSTM-RMLC approach with ILSVRC-trained weight initialization achieves the highest accuracy performance. The approach using the Softmax network and RML-CNN performs well and comes in second place. The suggested technique significantly outperforms all previous methods and improves accuracy by roughly 8.3% over the WARP [31] method and 4.4% over the SoftMax method [31]. The accuracy of methods with random weight initialization was much lower in all of the losses that were looked at, which suggests that there aren't many examples of networks with such a deep structure.

Method	**Accuracy (%)**
DGL [27]	82.44 ± 0.3
Improved GAN [28]	81.37 ± 2.3
ALI [29]	82.01 ± 1.6
Ladder Network [30]	79.60 ± 0.5
RML-CNN (ILSVRC init.) [31]	89.24 ± 0.3
WARP (ILSVRC init.) [31]	85.41 ± 0.4
Softmax (ILSVRC init.) [31]	89.32 ± 0.4
RML-CNN (random init.) [31]	52.37 ± 2.4
WARP (random init.) [36]	50.52 ± 0.5
Softmax (random init.) [36]	51.46 ± 1.1
CNN-LSTM- RMLC (random init)(proposed)	60.26 ± 0.8
CNN-LSTM- RMLC (random init)(proposed)	93.74± 0.4

Table 1 Classification Accuracy Results on CIFAR-10 image dataset

4.3. STL-10 image dataset

The STL-10 dataset, like the CIFAR-10 dataset, is a single object picture dataset that contains images from ten different categories. However, the picture data available is of excellent perseverance (96 x 96). This data set includes 5,000 identified photos and 100,000 unlabeled images. For our proposed approach trial, we employ captioned photos. The dataset is organized into ten folds, each of which has 400 annotated photos. The test data set is made up of 1000 photos. The accuracy results represent the average accuracy of ten different folds. Table 2 shows the experimental accuracy results.

Method	Accuracy (%)
RML-CNN (ILSVRC init.) [31]	91.11 ± 0.3
WARP (ILSVRC init.) [31]	85.63 ± 0.6
Softmax (ILSVRC init.) [31]	91.34 ± 0.3
RML-CNN (random init.) [31]	40.00 ± 0.9
WARP (random init.) [31]	39.15 ± 0.7
Softmax (random init.) [31]	39.27 ± 1.7
Haeusser et al. [32]	81.00 ±−−
Huang et al. [33]	76.80 ± 0.3
CNN-LSTM- RMLC (random init)(proposed)	53.84 ± 1.6
CNN-LSTM- RMLC (ILSVRC init)(proposed)	95.56 ± 0.4

Table 2. Mean Classification Accuracy Results on STL-10 Image dataset

The suggested supervised technique CNN-LSTM-RMLC achieves higher accuracy, while the softmax and RML-CNN methods rank second and third, respectively. Our suggested strategy with ILSVRC-trained weight initialization outperforms random weight initialization in all assessed models. The WARP approach outperforms softmax and RML-CNN methods with weight initialization of ILSVRC, but its accuracy performance is comparable to 0

Method	P-C	R-C	F1-C	P-O	R-O	F1-O	mAP
WARP [1]	59.3	52.5	55.7	59.8	61.4	60.7	–
RML-CNN [31]	62.4	61.1	59.8	62.8	65.4	64.1	71.5
WARP [31]	60.7	59.8	58.5	61.5	64.1	62.8	63.2
CNN – RMLC [31]	64.9	62.0	61.5	64.1	66.8	65.5	75.2
CNN + RNN [34]	66.0	55.6	60.4	69.2	66.4	67.8	–
RLSD [35]	67.7	56.4	61.5	70.5	59.9	64.8	67.4
ResNet-101 [48]	65.3	62.6	61.3	64.1	66.8	65.4	75.2
CNN-LSTM- RMLC (proposed)	74.4	68.4	67.5	73.6	75.2	74.6	82.6

Table 3. Top-3 ranked labels accuracy performance obtained on MS-COCO image dataset on 80 classes

4.3.1. NUS-WIDE dataset

NUS-WIDE is a multiple object web image database [36] with 269,648 manually annotated Flickr photographs and 5018 descriptions. These image data were manually classified into 81 classes, on average, with 2.4 class labels per image. We used 161789 picture files for training and 40500 image files for testing.

Method	P-C	R-C	F1-C	P-O	R-O	F1-O	mAP
WARP [1]	31.7	35.6	33.5	48.6	60.5	53.9	–
CNN – RMLC [31]	48.2	56.0	48.8	55.9	69.0	61.8	58.8
RML-CNN [31]	44.3	54.8	45.3	55.7	69.0	61.7	57.4
WARP [31]	40.2	51.8	42.2	53.9	66.7	59.7	48.5
Softmax [31]	31.7	31.2	31.4	47.8	59.5	53.0	–
CNN + RNN [34]	40.5	30.4	34.7	49.9	61.7	55.2	–
RLSD [35]	44.4	49.6	46.9	54.4	67.6	60.3	54.1
ResNet-107 (binary relevance) [48]	46.7	56.8	46.9	55.9	69.2	61.8	59.5
ResNet-101 (binary relevance) [48]	46.4	55.3	47.0	55.9	69.2	61.8	60.1
CNN-LSTM- RMLC (proposed)	54.6	62.3	55.6	62.2	73.4	64.7	63.4

Table 4. Top-3 ranked labels accuracy performance obtained on NUS-WIDE image dataset on 81 classes

Table 4. displays the findings of the experiment. Because WARP and RML-CNN share a similar network design, they may be directly compared. For multiple object image label classification, the CNN-RMLC technique produces less results than the ResNet-101 structure combined with a binary relevance cost function [48]. We also conducted novel tests using CNN feature maps from the trained approach proposed in [20]. In our experiment, we treated the CNN feature maps as manually extracted features and used LSTM and the loss function specified in equation-1 to simulate our proposed categorization. The CNN-LSTM-RMLC classifier approach achieves the highest P-C, F1-C, P-O, F1-O, and mAP scores. When compared to the results of CNN-RMLC [31], its mAP is 4.6% greater. It also outperforms the WARP approach greatly. Without leveraging bounding box approaches used for multiple object images, the suggested method outperforms the RSLD and CNN+RMLC methods by a slight margin.

4.3.2. PACAL VOC dataset

The PASCAL VOC [39] standard dataset is commonly used for recognizing multiple item labels. It features 9963 images in all. We used 6000 photos for training and 3963 images for testing in our research. The CNN-LSTM-RMLC method's performance is evaluated using the mean average precision (mAP) measure.

Table 5. Compares the experimental results achieved with different existing approaches to multiple object picture label classification. CNN-SVM [38] employs a support vector machine model trained on ImageNet single object image dataset feature maps. HCP [5] improves the feature maps of CNN, which was previously trained on the ImageNet dataset, by using information about regions of objects placed in an image, and outperforms approaches that do not employ object placement information. Our suggested method, CNN-LSTM-RMLC, does better than both the RARL [42] and HCP [5] methods by 1.2 mAP, even though the LSTM model doesn't take location information into account when classifying objects.

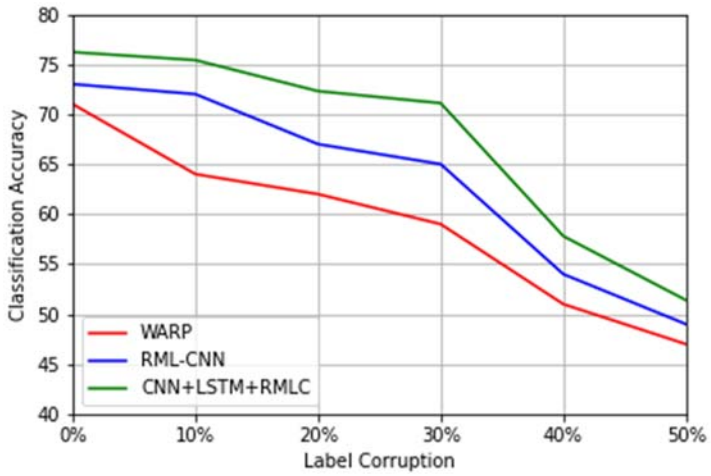
Method	plane	Bike	bird	boat	bottle	bus	car	cat	Chair	cow	table	Dog	horse	motor	person	plant	sheep	Sofa	train	Tv	mAP
HCP [5]	98.6	97.1	98.0	95.6	75.3	94.7	95.8	97.3	73.1	90.2	80.0	97.3	96.1	94.9	96.3	78.3	94.7	76.2	97.9	91.5	90.9
CNN-RNN [34]	96.7	83.1	94.1	92.8	61.2	82.1	89.1	94.2	64.2	83.6	70.0	92.4	91.7	84.2	93.7	59.8	93.2	75.3	99.7	78.6	84.0
RLSD [34]	96.4	92.7	93.8	94.1	71.2	92.5	94.2	95.7	74.3	90.0	74.2	95.4	96.2	92.1	97.9	66.9	93.5	73.7	97.5	87.6	88.5
CNN-SVM [38]	88.5	81.0	83.5	82.0	42.0	72.5	85.3	81.6	59.9	58.5	66.5	77.8	81.8	78.8	90.2	54.8	71.1	62.6	87.2	71.8	73.9
VeryDeep [39]	98.9	95.0	96.8	95.4	69.7	90.4	93.5	96.0	74.2	86.6	87.8	96.0	96.3	93.1	97.2	70.0	92.1	80.3	98.1	87.0	89.7
RDAR [40]	98.6	97.4	96.3	96.2	75.2	92.4	96.5	97.1	76.5	92.0	87.7	96.8	97.5	93.8	98.5	81.6	93.7	82.8	98.6	89.3	91.9
RARL [40]	98.6	97.1	97.1	95.5	75.6	92.8	96.8	97.3	78.3	92.2	87.6	96.9	96.9	93.6	98.5	81.6	93.1	83.2	98.5	89.3	92.0
[42]	99.9	98.4	97.8	98.8	81.2	93.7	97.1	98.4	82.7	94.6	87.1	98.1	97.6	96.2	98.8	83.2	96.2	84.7	99.1	93.5	93.8
CNN-LSTM-RMLC (proposed)	98.4	93.8	98.3	98.2	86.5	93.9	97.6	96.8	85.1	94.0	88.2	98.0	98.4	97.3	99.2	83.6	97.6	83.9	99.6	93.2	94.6

Table 5. Comparison of mAP on PASCAL VOC image dataset

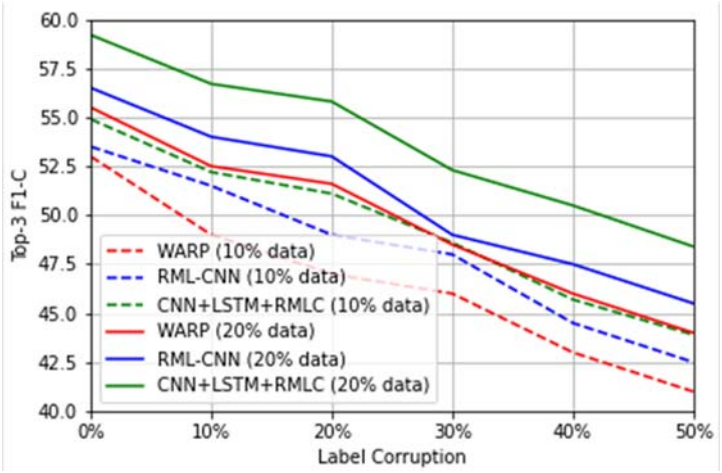
4.4. Experimental results on noisy data

We conducted an experiment using purposely altered image labels to demonstrate the usefulness of the suggested technique. We used 4000 tagged picture samples from the CIFAR-10 dataset in 5 batches in this experiment. To conduct tests with numerous parameters, subsampling of the MS-COCO and NUS-WIDE datasets was performed while maintaining the class label balance. To make the dataset noisy, the labels in a randomly selected subset are updated stochastically, ensuring that the original dataset contains balanced class label occurrences. The corresponding subset of class labels for all parameters is gradually increased from 10% to 50% to ensure consistency. The hyper parameters for the displayed metrics are enhanced, and the models are initialized with Image Net pre-trained model weights.

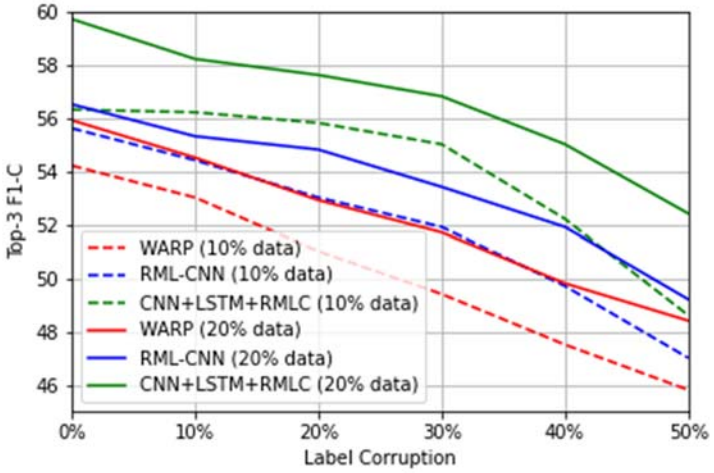
Figure 6 depicts the experimental results achieved. Across all datasets and parameters, our proposed model outperforms the RML-CNN and WARP models. Experiments using 10% and 20% of the MS-COCO and NUS-WIDE datasets yielded similar results. The CNN+LSTM+RMLC model's performance falls linearly as the fraction of degraded class labels increases. The resulting graph also shows that when more image samples are used (20% of the dataset instead of 10%), the results are better for all parameters.



(a) CIFAR-10



(b) MS-COCO



(c) NUS-WIDE

Fig 6.(a-c).Accuracy performance of noisy datasets

4.5. Qualitative results

Consider Figure 7 for a qualitative study of the findings obtained by the proposed model, which we will explain. The photos exhibited are from MS-COCO multiple object image dataset test samples. The truth labels come from the annotations that were added to the dataset. The projected labels, on the other hand, are the class labels for which our suggested model gave a high probability value.

(a) **True Labels**: Chair, TV, cup, table
Predicted Labels: Chair, TV, cup table

(b) **True Labels:** Motor bike, cat
Predicted Labels: Person, Motor bike, cat

(c)**True Labels:** person, baseball glove, baseball bat
Predicted Labels: person, baseball glove, sports ball, baseball bat

(d) (e) (f)

Fig.7. (a-c)Images from MS-COCO dataset with true labels and predicted label to depict the classification performance, (d-f) heat map visualization of superimposed saliency maps and original images.

The first column in Figure 7(a) illustrates that the proposed model accurately predicts all of the object names in the scenario. Figure 7(b) in the second column depicts the occurrence of inaccuracies due to dataset class inequity. Collecting multiple object image label datasets with an equal number of photos in each class is quite difficult. The suggested approach leans toward assigning high probability scores to often appearing class labels, such as the human label in the example image. We cannot, however, assume that the results of object classification are completely independent of the circumstances. The column in Figure 7(c.3) depicts significant correlations between class labels. Figure 7(c), for example, recognizes a sports ball solely because a picture sample has a person carrying a baseball bat and glove. This result demonstrates that the LSTM model can learn correlations between class labels in an image by utilizing feature maps recovered by CNN from the input image, namely features mined from the baseball bat and glove, which enhance the classification probability score for sports ball.

5. Conclusion and future work

For multiple item image label classification datasets with noisy data, we developed a robust supervised CNN-LSTM deep neural network architecture with a ramp loss function. Using LSTM in embedded vector space, this method combines the advantages of CNN feature extraction, label embedding, and objects label co-existence dependency for multi-label picture data classification. To maximize the margin, we employed the robust ramp loss available in WARP in conjunction with another loss term. On noisy datasets, the suggested framework significantly outperforms CNN-RMLC and WARP. Multiple object image label data sources for label classification, such as MS-COCO, NUS-WIDE, and PASCAL VOC, as well as single object label data sources, such as CIFAR-10 and STL-10, gave us the same level of accuracy. The conclusions of this paper are listed below:

1) The LSTM learning method can efficiently learn extremely intricate and associated image label data because the original image data are fully abstracted following the deep convolutional neural network's feature extraction procedure.
2) When compared to existing methods such as WARP, softmax, and RML-CNN, relative experiments on single object image data such as CIFAR-10 and STL-10 and multi-label image data such as MS-COCO, NUS-WIDE, and PASCAL VOC, CNN-LSTM-RMLC can improve accuracy performance by 4% and mean average precision (mAP) by 7.5.

The proposed framework of this paper does have some limits.

- The classification performance of the framework mainly relies on the image feature extraction of the CNN, which is comparatively inert at learning abstract features.
- Smaller objects in different parts of an image are still hard to spot because the overall visual features aren't good at telling them apart.

We intend to improve the presented framework solution in the future by employing efficient techniques such as a visual attention network to recognize numerous tiny items in a given image. In the meantime, we aim to investigate the multi-label classification problem in other contexts, e.g., using semi-supervised techniques to deal with largely available unlabeled image data and also some challenging data such as videos.

6. Conflicts of Interest

The authors have no conflicts of interest to declare.

References

[1] Gong, Y., Jia, Y., Leung, T., Toshev, A. and Ioffe. S. (2014): Deep convolutional ranking for multi-label image annotation. International Conference on Learning Representations, arXiv: 1312.4894v2 [cs.CV], 1-9.
[2] Guillaumin, M., Mensink, T., Verbeek, J., Schmid, C., and Tagprop. (2009): Discriminative metric learning in nearest neighbor models for image auto-annotation. IEEE 12th International Conference on Computer Vision, IEEE Xplore, pages 309–316.

[3] Xue, X., Zhang, W., Zhang, J., Wu, B., Fan, J., and Lu, Y. (2011): Correlative multi-label multi-instance image annotation. IEEE International Conference on Computer Vision (ICCV), IEEE Xplore, pages 651–658.
[4] Li, X., Zhao, F. and Guo, Y. (2014): Multi-label image classification with a probabilistic label enhancement model. Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, ACM Digital Library, 430–439.
[5] Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., and Yan, S. (2014): Cnn: Single-label to multi-label. arXiv:1406.5726, 6, 1-4.
[6] Deng, J., Dong, W., Socher, R., Li, J., Li, K. and Fei-Fei, L. (2009): Imagenet: A large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Xplore, pages 248–255.
[7] Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z. and Lu, H. (2019): Dual attention network for scene segmentation. IEEE Conferene on Computer Vision and .Pattern Recognition (CVPR), IEEE Xplore, pp. 3146-3154.
[8] Zhang, C.-L., Liu, X.-X. and Wu, J. (2019): Towards real-time action recognition on mobile devices using deep models. *arXiv:1906.07052*. [Online]. Available: http://arxiv.org/abs/1906.07052.
[9] Yu, W.-J., Chen, Z.-D., Luo, X., Liu, W. and Xu, X.-S. (2019): DELTA: A deep dual-stream network for multi-label image classification. *Pattern Recognition*, Elsevier, vol. 91, pp. 322-33.
[10] Hu, J., Shen, L. and Sun, G. (2018): Squeeze-and-Excitation networks. IEEE Conference on Computer Vision and .Pattern Recognition (CVPR), IEEE Xplore, pp. 7132-7141.
[11] Woo, S., Park, J. Lee, J.-Y. and Kweon, I.S. (2018): Cbam: Convolutional block attention module. In Proceedings of European Conference on Computer Vision (ECCV), pp. 3 -19.
[12] Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y. and Liu, W. (2019): CCNet: Criss-cross attention for semantic segmentation. In Proceedings of International Conference on Computer Vision (ICCV), pp. 603-612.
[13] Sumbul, Gencer and Demir, Begum. (2020): A Deep Multi-Attention Driven Approach for Multi-Label Remote Sensing Image Classification. IEEE Access, vol.8, 95934-95946.
[14] Li, P., Chen, P., Xie, Y. and Zhang, D. (2020): Bi-Modal Learning With Channel-Wise Attention for Multi-Label Image Classification. IEEE Access, vol.8, 65-77.
[15] Chen, B., Li, J., Lu, G., Yu, H. and Zhang, D. (2020): Label Co-Occurrence Learning With Graph Convolutional Networks for Multi-Label Chest X-Ray Image Classification. IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 8, 2292 – 2302.
[16] Park, J.-Y., Hwang, Y., Le, D. and Kim, J.-H. (2020): MarsNet: Multi-Label Classification Network for Images of Various Sizes. IEEE Access, vol.8, 21832-21846.
[17] Chen, B., Zhang, Z., Li, Y., Lu, G. and Zhang, D. (2020): Multi-Label Chest X-Ray Image Classification via Semantic Similarity Graph Embedding. IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 4, 2455 – 2468.
[18] Dong, N., Kampffmeyer, M., Liang, X., Wang, Z., Dai, W. and Xing, E. P. (2018): Unsupervised domain adaptation for automatic estimation of cardiothoracic ratio. In Medical Image Computing and Computer Assisted Intervention. New York, NY, USA: Springer, pp. 544–552.
[19] Diaz-Pinto, Colomer, A., Naranjo, V., Morales, S., Xu, Y. and Frangi, A. F. (2019): Retinal image synthesis and semi-supervised learning for glaucoma assessment. IEEE Trans. Medical Imaging, vol. 38, no. 9, pp. 2211–2218.
[20] Miyato, T. Maeda, S.-I., Koyama, M. and Ishii, S. (2019): Virtual adversarial training: A regularization method for supervised and semi-supervised learning. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 41, no. 8, pp. 1979–1993.
[21] Aviles-Rivero, Papadakis, N., Li, R., Alsaleh, S. M., Tan, R. T. and Schonlieb, C.-B. (2019): When labelled data hurts: Deep semi-supervised classification with the graph 1-Laplacian. *arXiv:1906.08635*. [Online]. Available: http://arxiv.org/abs/ 1906.08635.
[22] Iscen, A., Tolias, G., Avrithis, Y., Chum**,** O. (2019): Label Propagation for Deep Semi-supervised Learning**.** In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Xplore, 10.1109/CVPR.2019.00521.
[23] Wang, L., Liu, Y., Di, H., Qin, C., Sun, G. and Fu, Y. (2021): Semi-Supervised Dual Relation Learning for Multi-Label Classification. IEEE Transactions on Image Processing, Vol. 30, 9125 – 9135.
[24] Pascanu, R., Mikolov, T. and Bengio, Y. (2013): On the difficulty of training recurrent neural networks. International Conference on International Conference on Machine Learning. Vol.28, Pages 1310-1318.
[25] Dahl, G. E., Sainath, T. N. and Hinton, G. E. (2013): Improving deep neural networks for lvcsr using rectified linear units and dropout. International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE Xplore, pages 8609–8613.
[26] Turpin and Scholer, F. (2006): User performance versus precision measures for simple search tasks. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 11–18, ACM.
[27] Wang, G., Xie, X., Lai, J. and Zhuo, J. (2017): Deep growing learning, IEEE International Conference on Computer Vision (ICCV), Volume: 1, Pages: 2831-2839.
[28] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. and Chen, X. (2016): Improved techniques for training gans, Proceedings of the 30th International Conference on Neural Information Processing Systems, Pages 2234–2242.
[29] Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., L.abd, A., Arjovsky, M. and Courville, A. (2017): Adversarially learned inference, *International Conference on Learning Representations (ICLR),* 1-18.
[30] Rasmus, A., Valpoa, H., Honkala, M., Berglund, M. and Raiko, T. (2015): Semi-supervised learning with ladder networks, In Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS), Volume 2, Pages 3546–3554.
[31] Hakan Cevikalp, BurakBenligiray, OmerNezihGerek. (2020): Semi-supervised robust deep neural networks for multi-label image classification, Pattern Recognition, Elsevier, vol.100.
[32] Haeusser, P., Mordvintsev, A. and Cremers, D. (2017): Learning by association a versatile semi-supervised training method for neural networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 10.1109/CVPR.2017.74.
[33] Huang, C., Loy, C.C. and Tang, X. (2016): Unsupervised learning of discriminative attributes and visual representations. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 10.1109/CVPR.2016.559.
[34] Wang, Jiang & Yang, Yi & Mao, Junhua & Huang, Zhiheng & Huang, Chang & Xu, Wei. (2016): CNN-RNN: A Unified Framework for Multi-label Image Classification. 2285-2294. 10.1109/CVPR.2016.251.
[35] Zhang, J., Wu, Q., Shen, C., Zhang, J. and Lu, J. (2018): Multi-label image classification with regional latent semantic dependencies. IEEE Transactions on Multimedia, Vol. 20, No. 10, 2801 – 2813.
[36] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C. and Fei-Fei, L. (2015): ImageNet large scale visual recognition challenge. International Journal of Computer Vision, Vol.115, 211–252.
[37] Everingham, M., Van Gool, L., Williams, C. K., Winn, J. and Zisserman, A. (2010): The Pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303–338.
[38] Razavian, A.S., Azizpour, H., Sullivan, J. and Carlson, S. (2014): CNN features off-the-shelf: An astounding baseline for recognition. In Proceeds of IEEE Conference on Computer Vision Pattern Recognition Workshops, pp. 806–813.
[39] Simonyan, K. and Zisserman, A. (2014): Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.
[40] Wang, Z., Chen, T., Li, G., Xu, R. and Lin, L. (2017): Multi-label image recognition by recurrently discovering attentional regions. In Proceedings of IEEE International Conference on Computer Vision, pp. 464–472, arXiv:1711.02816v1 [cs.CV].

[41] Joseph James, S., Lakshmi, C. (2020): A Study: Multiple-Label Image Classification using Deep Convolutional Neural Network Architectures. Advances in Intelligent Systems and Computing, Springer, Vol. 1059, pp.759-773.
[42] Wen, S., Liu, W., Yang, Y. and Zhou, P. (2020): Multi-label Image Classification via Feature/Label Co-Projection. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2168-2216.
[43] Lakshmi, C. and Sundararajan, M. (2009): Biometric Security system using Face Recognition. International Journal of Pattern Recognition and Artificial Intelligence, pp. 125-134.
[44] Karpathy, Andrej & Li, Fei. (2015): Deep visual-semantic alignments for generating image descriptions. 3128-3137. 10.1109/CVPR.2015.7298932.
[45] Harzallah, H., Jurie, F. and Schmid, C. (2009): Combining efficient object localization and image classification. IEEE 12th International Conference on Computer Vision, pp. 237–244, 10.1109/ICCV.2009.5459257.
[46] Cao, J., Wu, C., Chen, L., Cui, H. and Feng, G. (2019): An Improved Convolutional Neural Network Algorithm and Its Application in Multilabel Image Labeling. Computational Intelligence and Neuroscience, Vol.2019, 1-12, 10.1155/2019/2060796.
[47] Yeh, C.-K., Wu, W.-C., Ko, W.-J. and Wang, Y.-C. F. (2017): Learning deep latent space for multi-label classification. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17) pp. 2838–2844.
[48] Zhu, F., Li, H., Ouyang, W., Yu, N. and Wang, X. (2017): Learning Spatial Regularization with Image-Level Supervisions for Multi-label Image Classification, pp. 2027-2036. 10.1109/CVPR.2017.219.

Authors Profile

Dr. C. Lakshmi holds Ph.D in Computer Science and Engineering from SRM University and is Professor in the School of Computing, SRM Institute of Science and Technology, deemed to be a university in Kattankulathur Campus, Chennai. Her main area of research interest includes pattern Recognition, image processing, machine learning, software Engineering and web services. She has published patents on “Object Detection and Labeling in an image", A System for Preventing Road Accidents", A System and a method for Protecting Crop fields from Wild Animals” and few more; She has served as a Resource person in many Faculty Development Programs and workshops; PI in numerous Intramural & Extramural funded research projects. She is a Life time member of the Indian Society for Technical Education (ISTE), besides the author is a member of many professional bodies, including the Institution of Engineers (IEI), the Association of Computing Machinery (ACM), and the Indian Science Congress Association (ISCA), International Association of Computer Science and Information Technology, Singapore, International Association of Engineers-IAENG, Hong Kong. She has published several papers in well-known peer-reviewed journals.

Joseph James S holds a M.E degree in Software Engineering from Anna University and is a research scholar in the department of Computer Science and Engineering, SRM Institute of Science and Technology, deemed to be a university in Kattankulathur Campus, Chennai. His main area of research interest includes pattern Recognition, image processing, machine learning and deep learning. He has published patents on”Object Detection and Labeling in an image", A System and a method for Protecting Crop fields a from Wild Animals”; served as a Resource person in many Faculty Development Programs and workshops, Co-PI in numerous Intramural & Extramural funded research projects , published and presented various Research paper at numerous National & International Platform. Besides, the author is a member of many professional bodies, including the Institution of Engineers (IEI), the Association of Computing Machinery (ACM), and the Indian Science Congress Association (ISCA).He is member of IEEE and Indian Science Congress. He has published several papers in well-known peer-reviewed journals.