IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 15, 2021, accepted December 30, 2021, date of publication January 6, 2022, date of current version January 20, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3140861

An Improved Automatic Image Annotation Approach Using Convolutional Neural Network-Slantlet Transform

MYASAR MUNDHER ADNAN[1,2], MOHD SHAFRY MOHD RAHIM[1], AMJAD REHMAN KHAN[3], (Senior Member, IEEE), TANZILA SABA[3], (Senior Member, IEEE), SULIMAN MOHAMED FATI[3], (Senior Member, IEEE), AND SAEED ALI BAHAJ[4]

[1]School of Computing, Faculty of Engineering, University Technology of Malaysia, Johor Bahru 81310, Malaysia
[2]Department of Islamic University, Najaf 54001, Iraq
[3]Artificial Intelligence and Data Analytics Laboratory, College of Computer and Information Sciences (CCIS), Prince Sultan University, Riyadh 11586, Saudi Arabia
[4]MIS Department, College of Business Administration, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia

Corresponding author: Amjad Rehman Khan (arkhan@psu.edu.sa)

This work was supported by the Artificial Intelligence and Data Analytics Laboratory (AIDA), College of Computer and Information Sciences (CCIS), Prince Sultan University, Riyadh, Saudi Arabia.

ABSTRACT Every day, websites and personal archives generate an increasing number of photographs. The extent of these archives is unfathomable. The ease of usage of these enormous digital image collections contributes to their popularity. However, not all of these databases provide appropriate indexing data. As a result, it's tough to find information that the user is interested in. Thus, in order to find information about an image, it is necessary to classify its content in a meaningful way. Image annotation is one of the most difficult issues in computer vision and multimedia research. The objective is to convert an image into a single or numerous labels. This necessitates a grasp of the visual content of an image. The necessity for unambiguous information to build semantic-level concepts from raw image pixels is one of the challenges of image annotation. Unlike text annotation, where a dictionary links words to their meaning, raw picture pixels are insufficient to construct semantic-level notions directly. A simple syntax, on the other hand, is well specified for combining letters to form words and words to form sentences. The automatic feature extraction for automatic annotation was the emphasis of this paper. And they employed a deep learning convolutional neural network to build and improve image coding and annotation capabilities. Performance of the suggested technique on the Corel-5K, ESP-Game, and IAPRTC-12 datasets. Finally, experimental findings on three data sets were used to demonstrate the usefulness of this model for image annotation.

INDEX TERMS Automatic image annotation, deep learning, features extraction, slantlet transform, technological development.

I. INTRODUCTION

In recent years, it has become very difficult to search an image in a large image database. Many methods have been proposed to access an image [1], [2]. Low-level visual content such as shape, color, and texture, as well as labels or keywords that convey the semantic meaning of the provided image, can be used to retrieve the image. The user must provide an input image of a query to access photos using low-level visual functions, and the search result returns a set of images that are visually similar to the query image. However, many customers find it challenging to find a query image that matches their needs every time. CBIR (Content-Based Image Retrieval) is a technique for recovering images from low-level visual attributes. Another way for overcoming the issues of CBIR systems is to assign labels to all photos in the database. These can be found using these labels [3]. The key advantage of this method is that the image can be retrieved in the same way that a text document can be retrieved. This label assignment method is called image annotation. During the last decade, there have been significant breakthroughs in the field of computer vision; using computers to solve

The associate editor coordinating the review of this manuscript and approving it for publication was Geng-Ming Jiang.

problems involving the identification and classification of images has never been easier. Because of this, an alternative to the existing manual process is now possible: one that uses a deep learning neural network model to classify and tag images to automate this process. However, as training deep neural networks from scratch is time-consuming, a computationally cheaper alternative was desirable. Thus, the goal of the project became to design and implement such a solution—something that could categorize images in reasonable amounts of time even on large data sets, with limited computing power, but could still achieve reasonably accurate results. Furthermore, additional resources could be allocated to the project for more complex deep learning systems if results were good. A deep neural network, such as those used in deep learning, works in much the same way as the neural structure of the human brain—neurons connected, with the outputs of neurons further along in the network relying on the outputs of earlier neurons as their inputs. Teaching a deep neural network is best done through what is commonly referred to as ''supervised learning''—giving the network a batch of data and its associated labels, and letting the network attempt to process the data. By comparing the network's actual output to the expected output that the provided data labels contain, the network's activation probabilities can be adjusted to make the actual output match the expected output. The novelty of this study is as follows:

1. Combining multiple features (SLT, YCbCr, LBP) based on CNN features and neighbors to achieve a balance between precision and recall by selecting CNN with Slantlet Transform. Flexible annotation and improved accuracy are achieved.

2. A word2vec model with CNN-SLT was used to predict the image annotation using both word2vec distributed representation and learning representation. The distributed representation approach included the encoding and storage of information regarding the image features.

3 - Based on our proposed as some image annotation models require considerable computation time and complexity during the training phase, they become computationally intensive when training datasets are large. The proposed method is efficient in terms of computing time.

The rest of this paper is organized as follows. In the following section, we briefly introduce the image annotation. The remainder of this manuscript contains, sections 2 introduce some important background knowledge in the form of related works, Section 3 illustrates an improved deep feature extraction method, while section 4 proposes a novel method for image annotation. Section 5, we compare and analyze our proposed method with numerous methods such as MBRM [4], SEM [5], FastTag [6], and 2PKNN [7]. Finally, section 6 our conclusion summarizes the current research and suggests possible research venues for the future.

II. CNN RELATED WORKS

An automatic image annotation system involves assigning keywords from a dictionary to an image. Thus, input is the target image, and output is the best description of that image in terms of keywords. A computer can easily measure color, texture, and shape, but they cannot be interpreted semantically, unlike people who can easily deduce meaning from images. Thus, an essential challenge in automatic image annotation is to bridge the semantic gap between low-level computer features and the interpretation of images by humans [8], [7]. Several approaches have been proposed to address the issue of automatic image annotation in recent years. Several different models can be used to describe these approaches. There are three main models in automatic annotation: graphic models, generative models, and discriminating models. [9] recurrent neural networks (RNNs) and deep convolutional neural networks (MTCs) in a unified setting address the dependencies between labels in images. In the proposed CNN-RNN framework, label-image relationships characterize both semantic label dependencies and correlations between image labels. CNN generates an image's vectors. A multi-label prediction can be calculated sequentially using NRNs based on the vector of the image characteristics and outputs of recurrent neurons, where the prior probability of a label can be calculated for each step. Figure 1 illustrates the general pattern of this method.

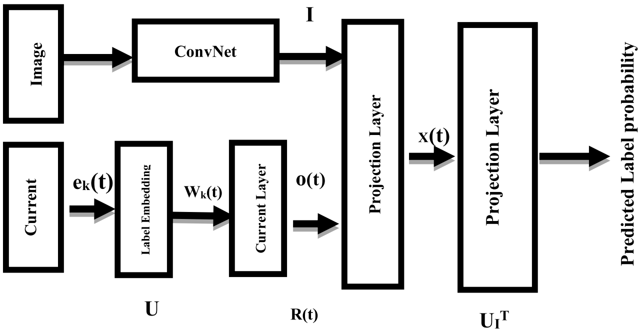


FIGURE 1. General method scheme [9].

According to (Murthy, Maji, and Manmath [7]), the authors used CNN features and word representation vectors to perform image annotations. Canonical correlation analysis (CCA) is the basis of the proposed model that helps model both visual and textual functions simultaneously. Recurrent neural networks are used to determine the visual functions of an image. Word2vec architecture is being used to remove textual functions [10]. By late 1990s, this system was already reading almost 10% of all the cheques circulated in the United States. Later, Microsoft deployed many optical handwriting recognition and character recognition systems using CNNs [11]. As an experiment in the early 1990s, CNNs were used to detect objects in natural images, including hands and faces [12], [13]. In the 1990s, convolutional network was employed to solve issues with speech recognition [14] and document reading [15], [16], while time-delay neural networks were employed for extracting meaningful content. A hybrid of a probabilistic model with CNN was used in document reading for barriers that exist in languages. In the United States, this application was employed extensively to read cheques. Meanwhile, Tao *et al.* A deep learning approach based on transfer learning and multiple tasks learning was proposed for analyzing images of biological components [17], [18]. On the other hand, [19] put forward a DL algorithm based on CNN along with reported results exceeding the existing ML strategies. In the visual recognition challenge, the proposed work won accolades for the researchers. Learning and modeling complex relationships can be done with artificial neural networks. However, choosing the number of hidden layers and the number of neurons in each layer presents a problem. In fact, the exact relationship between entry and exit can be challenging to explain.

III. THE ARCHITECTURES OF THE CNN

Convolutional neural networks (CNN) are artificial neural networks used to extract local features from data. CNN simplifies the network model by allocating weights to singular features, thereby lowering the overall weights CNN has become widely popular in the field of pattern recognition due to its unique characteristics [20]. For example, a CNN is employed by the document reading system trained jointly alongside a probabilistic model comprising language constraints, In CNN architecture, there are three key constituents or layers: 1) input, 2) hidden, and 3) latent. One may categories these latent (hidden) layers as either a pooling layer, fully-connected layer, or convolutional layer. Figure 2 shows these layers adapted from [21].

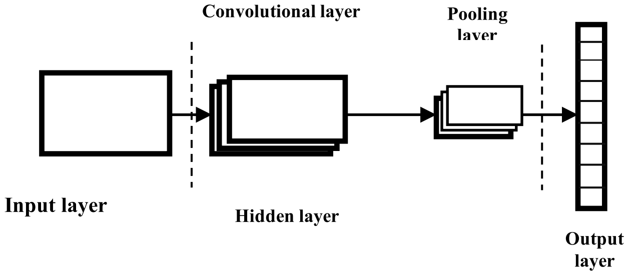


FIGURE 2. The pipeline of the general CNN architecture (Liu and an, 2018 [21]).

A. CONVOLUTIONAL LAYER

CNN architecture includes a convolutional layer as its primary layer. Convolution involves iteratively applying a function to a varying function and then evaluating its output [22]. This layer is made up of several maps of neurons, this is also known as filters or features maps. According to size, it is relatively identical to the input data's dimensionality. One can also interpret neural reactivity through the quantification of discrete convolution of receptors. Activation functions and total neural weights of input are calculated during the quantification process. Figure 3 briefly demonstrates the discreet convolution layer.

B. MAX POOLING LAYER

The max pooling layer, several grids are created from the split convolution layer output. In matrices, the maximum values are sequenced [22]. Then, the average or maximum value of each matrix is calculated using operators. Figure 4 illustrates the procedures for maximum pooling.

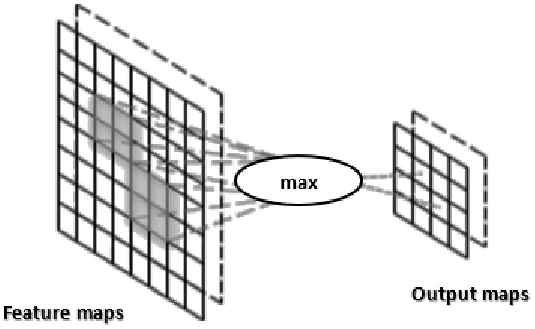


FIGURE 3. The illustrates discreet convolution layer [22].

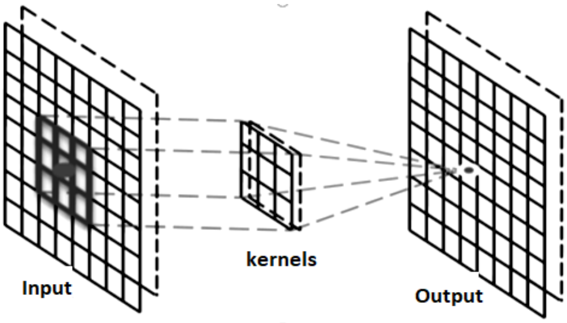


FIGURE 4. Illustrates max pooling procedures layer [22].

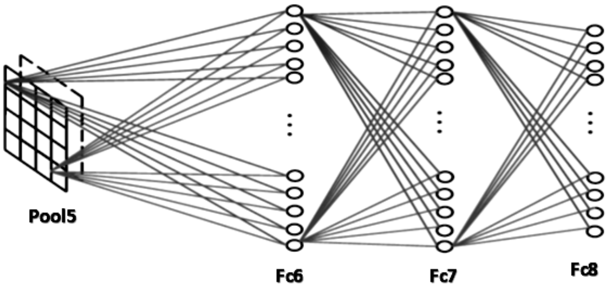


FIGURE 5. The illustrates discreet of the fully-connected layer [20].

C. FULL CONNECTION LAYER

Full connection layer refers to an almost complete CNN that comprises 90% of overall CNN architectural parameters. Input can be sent using predefined vector lengths across the network in this layer [20]. Figure 5 presents a brief illustration of the full connection layer. Dimensional data is transformed through layers before it can be classified. Furthermore, the convolutional layer is transformed to maintain the integrity of the information.

Fully-connected layers are connected to neurons from an earlier layer. As the final network layer, these fully connected layers assist in the classification process. An example of a CNN that explains all three layers is shown in Figure 6.

Due to its design for object recognition, CNN may not be the best solution for our problems. To improve performance, we will design a customized network structure according to

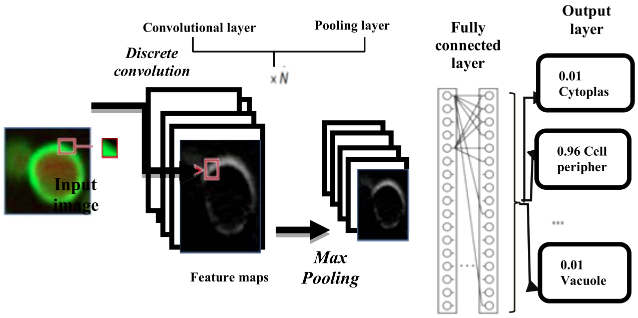


FIGURE 6. An example describing the complete CNN architecture.

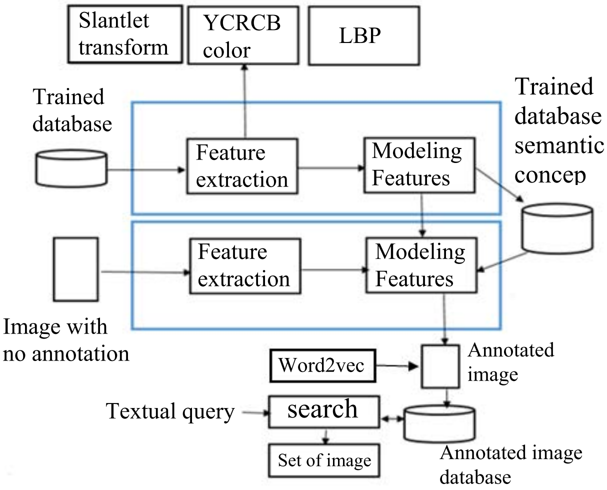


FIGURE 7. The illustrates discreet of the fully-connected layer.

the problem domain. We demonstrate that all of our methods set the bar high for performance on all of our problems through experiments.

IV. PROPOSED AIA ARCHITECTURE

The architecture proposed by initial investigation from the literature reviewed is discussed in this section. The proposed system consists of three major phases. Training is essential part of the system where a database of tagged images is used. The trained system then works on raw data to output the annotated image in the second phase. In the last phase image retrieval should be carried out to evaluate annotation results. The standard training database is used under an automatic features extraction using CNN in the first training phase. The automatic features extraction process gives the feature vector easily by understanding the contents of the images. Modelling of the features via learning mechanism is the next activity. It generates model for annotation that is to annotate new images.

In the second phase the un-annotated image is the input. Extraction of features is the next activity to generate visual characteristics of the contents to be applied to the annotation model trained in the previous phase. The model generatdevat an earlier phase will assign proper semantic labels to the image as per the contents. So, this will result in an annotated image as output. In the third phase the images from annotation phase are taken as data store. A textual query will be fired and the system will give list of appropriate images. Since the annotation is content based the retrieval of images will become easier and accurate. Figure 7 shows the framework of the proposed system architecture for the automatic image annotation.

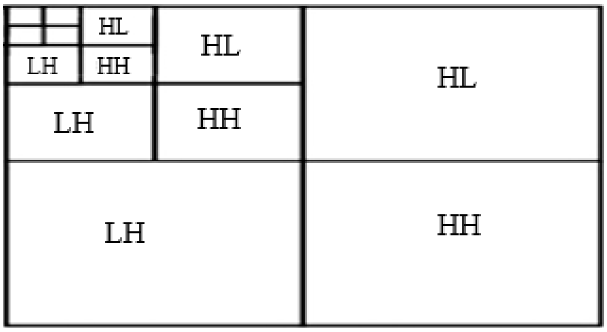


FIGURE 8. 4-Level wavelet transformation of an image.

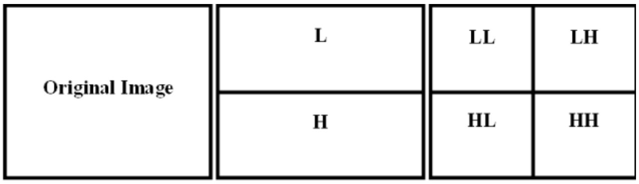


FIGURE 9. The conventional 2D SLT decomposition schemes for dividing an image.

This section describes the Deep learning technology and the renowned Deep learning architecture Convolutional Neural Network CNN.

V. FEATURES EXTRACTION

There are several factors contributing to an automatic image annotation (AIA) process, such as feature extraction, identification of suitable features for use in the AIA, mathematical transforms selected for determining the feedback usage, etc. An effective annotation system complements these distinguishing factors. Researchers used the low-level and high-level information contained in an image such as texture, shape, and color to reconstruct the image. Automatic features extraction will be discussed to achieve the research objective, which is to implementation of new AIA system based on automatic features extraction and object learning representation and select the most adequate features, the first one is to extract shape using Slantlet Transform, second to extract color using YCbCr Colour Space and extract texture features using Local Binary Pattern (LBP). In what follow three types of features that can be used in our experiment.

A. WAVELET TRANSFORM

Wavelet techniques are used to remove noise from image or signal for data classification and data compression, which means wavelet can be used to perform various image and signal processing operations. However, wavelet technique has certain disadvantages: if there is shifting in time for input signal, there will be unpredictable changes in values of transform coefficients. Coefficients cannot discriminate between input signal shifts due to this shifting Discrete Wavelet Transform (DWT) [23] [24]. Furthermore, images contain different edges with various orientations and DWT can only support horizontal, diagonal and vertical orientations. So, wavelet has poor directionality. The major drawback of DWT is that it considers only real coefficient filters associated with real wavelets and gives only real-valued approximations, but complex signals can be used for various operations in image processing, and phase of the complex signal is calculated by its real and imaginary coefficients. Here DWT fails to provide accurate phase information. This disadvantage can be overcome by using complex valued filtering [25]. Figure 8 below shows 4-level DWT decomposition of input image.

VI. SLANTLET TRANSFORM

The SLantlet Transform (SLT) was an orthogonal Discrete Wavelet Transform (DWT) method with 2 zero moments and improved time localization. The SLT consists of all the usual features of the filter bank implementation but has a scale dilation factor of 2. This basis was not dependent on the iterated filter bank like DWT; however, different filters were used for every scale.

Generally, in the 2D SLT decomposition, the image is categorized into 4 components, LL (Low-Low), LH (Low-High), HL (High-Low), and HH (High-High) [26], as shown in Figure 9, where L and H signify the low and high frequency band, respectively. Each of them carries different image information. The low-frequency band component marked as LL of the image maintains the original image information. Conversely, the medium- and high-frequency bands, LH, HL, and HH carry the information related to the image's edge, contour, and other details. Therefore, high coefficients represent the critical information in the image. Meanwhile, the insignificant (small) coefficients are considered as noise or worthless information. Thus, these small coefficients must be ignored to get the best results in subsequent operations.

The SLT [27] process is a multi-resolution method suitable for piecewise linear data. SLT was an orthogonal DWT method having 2 zero moments and better time localization features. It is based on designing different filters for different scales unlike the iterated filters approach using DWT. Previously, SLT is used in awasariety of applications such as estimation, compression, fast algorithms and de-noising various input images. In parallel processing, SLT is implemented as a filter bank with parallel structures, where different filters are configured for different scales as opposed to filter iterations at different levels. Following [28], the coefficients of the filters are calculated using the SLT equations.

A. YCbCr COLOUR SPACE

Different colour models have shown varied visibility of the tampering traces. The image forgery detection methods generally use the RGB or grey-scale colour systems. Many recent studies [29]. noted that the use of the chromatic channels instead of RGB or luminance improved the detection performance. The YCbCr colour model represented the colours in the luminance (Y) and chrominance (Cb and Cr) components. Eq. (1) presents a formula that computes the Y, Cb and Cr channels using the R, G and B channels. Figure 10 depicts the Y, Cb and Cr channels in the colour image.

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.177 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix} \tag{1}$$

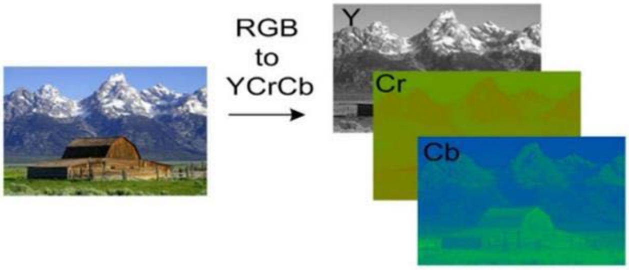


FIGURE 10. RGB image and its YCbCr counterpart.

B. LOCAL BINARY PATTERN (LBP)

LBP was a local operator that can discriminate between various texture types. The initial LBP operator [30]. defined the label (LBP code) of every pixel in the image. For computing this LBP code, the researchers compared the 3 × 3 neighborhood pixels with the central pixel value (threshold): It was seen that if the neighboring pixel values were lesser than the center value, it would hold the binary digit '0', or else, it would hold '1'. All the binary digits of the neighbors were concatenated for building the binary code. The LBP code was seen to be the decimal value of the binary code. The example shown in Figure 12 describes the LBP code computation method. LBP_(P, R) refers to the LBP operator, and was defined as:

$$LBP_{P,R} = \sum_{i=1}^{p-1} S(p_i - p_c)2^i \tag{2}$$

wherein; P refers to the no. of pixels in a neighborhood; R was the radius; P_c was a center pixel value; while the thresholding formula was defined as:

$$(\mathrm{P_i} - \mathrm{P_c}) = \begin{cases} 1 & \mathrm{P_i} - \mathrm{P_c} \geq 0 \\ 0 & \mathrm{P_i} - \mathrm{P_c} < 0 \end{cases} \tag{3}$$

In the LBP computation method, initially the T of the local 3 × 3 neighborhood of a Cb image was defined as a joint distribution of all grey-levels of 9 image pixels:

$$\mathrm{T} = \mathrm{P}\{g_0, g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, \}$$

wherein; g_i (i = 0, . . . , 8) corresponds to all grey values in the pixels present in a 3 × 3 neighborhood, based on

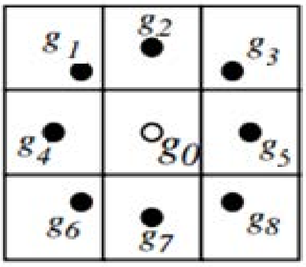


FIGURE 11. A circular symmetrical neighboring set of 8 pixels in the 3 × 3 neighborhood.

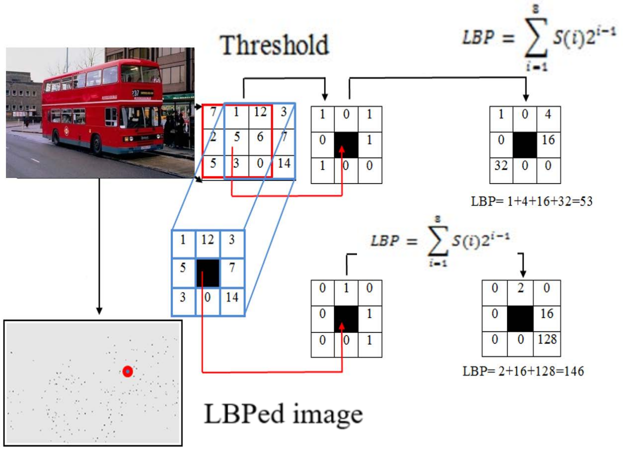


FIGURE 12. LBP image generation from the input image.

the spatial layout, described in Figure 11 A pattern of the neighbors is known as the "window" that slides over the complete image, pixel by pixel, from left to right until it reaches the final column. Thereafter, this window again goes to the 1st column and moves downwards from the top to bottom.

Based on this theoretical explanation, Figure 12 describes a method for computing the LBP.

This recherché suggests a novel way of using CBIR system using merging features extraction system based on CNN architecture, the proposed system combine and merge automatically between the features extracted.

VII. Word2vec REPRESENTATION

The Word2vec was seen to be a successful and popular natural language processing NLP approach for words representation [4]. This approach involved encoding and storage of information within the system by interacting with the other objects. The human memory structure inspired the distributed representation technique, wherein all memories are stored in a "content-addressable" manner. The content-based storage efficiently recalls all memories based on their partial description. Since these content-addressable thoughts and their properties are stored in a close proximity, the systems possess a viable infrastructure for generalizing the features for any item.

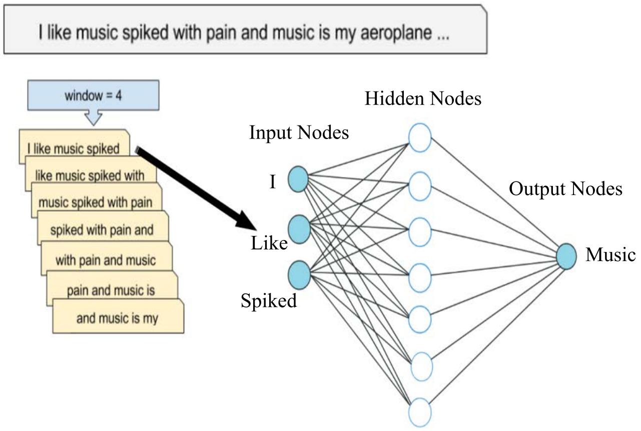


FIGURE 13. The distributed representation, WORD2VEC, used in natural language processing (NLP).

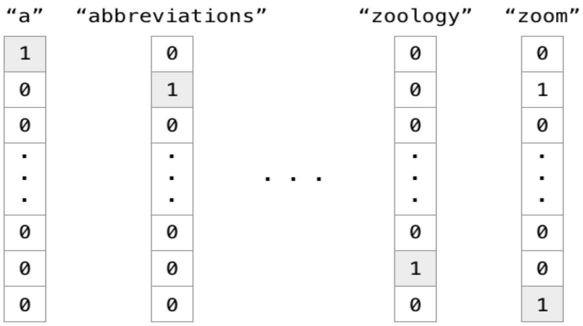


FIGURE 14. The Word2vec model.

The continuous vector representation, which acts like a distributed representation of words, was used in the Natural Language Processing (NLP) system for efficiently representing the semantic/syntactic units having multiple applications. Figure 13 illustrates the distributed representation, WORD2VEC, used in natural language processing (NLP).

A word representation is learned unsupervised by Word2vec [55]. It is necessary to feed these models a sufficiently large, properly encoded text. As shown in Figure 14, the main principle of word2vec is that a piece of text is given to the neural network, which is then divided into portions of a certain size (called windows). The network analyses every fragment as a pair of target words and contexts. Below is an example of a target word and context. The target word is "music" and the context is "I", "like", and "spiked".

Each, fragment's middle is used as the target word during such training while the rest is used as context. The Word2vec model learns word embedding's by predicting the middle word based on its context. Hidden layers of neural networks each contain a set of weights for each of the words (in the example above, 7 neurons). When a learning process is complete, the weights act as vectors representing the words. The important trick about word2vec is that we're not too concerned with the results of the neural network. At the end of the training phase, we extract the internal state of the hidden layer, resulting in a vector representation for every

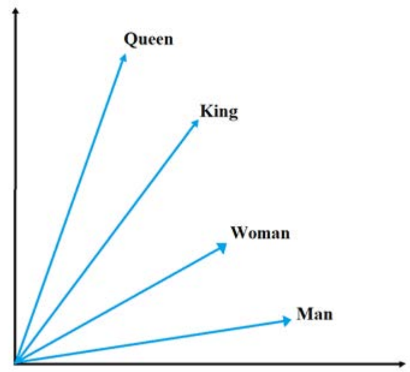


FIGURE 15. Word2Vec wherein every word was embedded in the vector in an n-dimensional space.

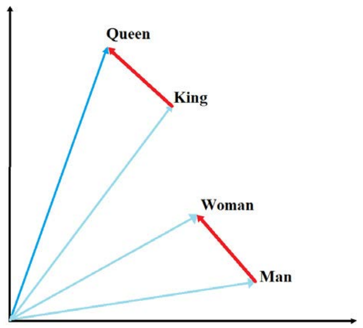


FIGURE 16. Word2Vec wherein the words with similar vector representations display multiple similarity degrees.

word. In Figure 14, a neural network is composed of an input layer, a hidden layer, and an output layer. There is no activation function in the hidden layer, and neurons combine weights and inputs linearly (multiply each input by its weight and add them up). In fact, word2vec requires that each word is represented as a hot encoded vector in the input layer. In the model, every word was embedded with the vector in the n-dimensional space. The similar words had closer vectors as described in Figure 15, like "King, Queen" and "Woman, Man", wherein the similarity was based on the syntax and semantics. These vectors were trained based on the idea that the meaning behind the words was characterized by their context, i.e., neighboring words. Hence, the various words and their context were considered the positive training samples [4], [46].

They observed exciting patterns by training the word vectors in the natural language. The words, having a similar vector representation, exhibit multiple similarity degrees. For example, Figure 16 shows that the words ensemble their closest vector with the word [7].

In this paper, the researcher predicted the image annotation using the word2vec distributed representation and learning representation as in the word2vec model. The distributed representation approach included the encoding and storage of information regarding the image features. The learning representation of objects in images uses the internal state of the hidden layer at the end of the training phase, which yields precisely one vector representation per object. In this section we describe the distributed and learning representation and how we can use both techniques to create new object or image representation.

VIII. EXPERIMENTAL RESULTS AND COMPARISON

By analyzing and comparing the reasons for choosing CNN-SLT, we analyze the quality of these models using three standard benchmark datasets. The CNN annotation framework is a comprehensive method for solving image annotation problems. To ensure that the annotation effects of the framework are optimal, the system combines and merges the features extracted with the CNN architecture. We begin with an explanation of the datasets and evaluation metrics. Secondly, the results of each method are presented and analyzed briefly. The final step in our analysis presents a comparison between our model and several state-of-the-art annotation methods. We also provide several examples of how the annotation process works.

A. DATASET

In our experiments, we used three popular image annotation databases: Corel-5K [31], ESP-Game [32] and IAPRTC-12 [33]. Corel-5K: This is the most popular base for annotation and image search. A vocabulary of 260 keywords is used for both training and testing the system, which has 4,500 images for training and 500 images for testing Images are categorized into 50 categories, each containing 100 images. Each image has 1 to 5 keywords manually annotated, with an average of 3.4 keywords per image. A subset of the 20770 images used in literature were obtained from the ESP-Game dataset [54]. With a vocabulary of 268 keywords, this subset consists of 18689 images for training and 2081 images for tests. Images are annotated with an average of 4.7 keywords per image.

IAPRTC-12: This database collects about 20,000 natural images. Its contents include 17665 training images and 1962 tests images, with a vocabulary of 291 keywords. The average number of keywords per image is 5.7. Table 1 shows detailed information about each database [56].

Table 2 represents some sample images with their annotations from the three databases used for the experimentation section. For example, the second image of the Corel-5k base represents an image annotated by the keywords: "sky," "jet" and "plane."

B. PERFORMANCE EVALUATION

Several quality measures for image annotation systems are used in the literature. According They can be divided, to Kwasnicka [34], Two main categories can be identified: measures by annotation and measures per word. In the following sections, we detail these two categories and the measures used in this study.

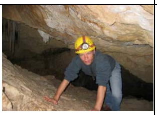
1) MEASURES BY ANNOTATION

Annotation measurements focus on the result of image-by-frame annotation. First, the measurements are calculated after

TABLE 1. Detailed information for each database.

database	Number of images	vocabulary size	Train size	Test size	Words per image	Images per word
Corel-5K	5 000	260	4 500	500	3.4	58.6
ESP-Game	20 770	268	18 689	2 081	4.7	362.7
IAPRTC-12	19 627	291	17 665	1 962	5.7	347.7

TABLE 2. Examples of images from test bases.

Corel-5k	 sky, sun, clouds, tree	 sky, jet, plane	 bear, polar, snow,
IAPRTC-12	 grandstand, lawn, player, roof, round, stadium	 helmet, jean, lamp, man, sweater	 city, house, roof, sky, valley, view
ESP-GAME	 round, stone, green, sky, grass, man, bunker, building, concrete	 stone, white, cow, dirt, tail, bull, grass,	 pink, silver

the annotation of each image. Following that, we calculate the average values for each image in the test set. [35].

2) ANNOTATION RATE

One of the most fundamental measures of quality for automatic annotation methods is the annotation rate. It measures the number of words predicted correctly in the annotation. If all words are correctly predicted, the measure has a value of 1, and if none of the words are correctly predicted, it has a value of 0. The average annotation rate is based on the arithmetic average of all test images [36].

$$\tau_i = \frac{c_i}{l_i}, \quad \tau = \frac{1}{I}\sum_{i=1}^{I}\tau_i \tag{4}$$

where τi the annotation rate of the image i, τ represents the average annotation rate of the test set, an annotation of image i represents the number of words correctly predicted, a length of the annotation represents the size of the test set, and ci represents the number of words correctly predicted [37].

3) STANDARDIZED SCORE

The standardized score is the second measure in this category. Again, it's rated by NS. The annotation rate is similar to it, but it also counts all misinterpreted words as a penalty.

$$NS_i = \frac{c_i}{l_i} - \frac{d_i}{V - l_i}, \quad NS = \frac{1}{I}\sum_{i=1}^{I} NS_i \tag{5}$$

where V represents the size of the vocabulary and di represents the number of words predicted incorrectly. The average standardized score is calculated on all annotations in the test set [38].

4) MEASURES PER WORD

It is possible to calculate the measurements per word when all words in the test set are annotated. Annotated images contain information, which is gathered by words. Then, for each word in the vocabulary, averages are calculated [39].

5) PRECISION AND RECALL

Suppose an e label is present m1 times in the images of the truth-ground, and appears in m2 images during tests from which m3 predictions are correct.

Precision: the relationship between images that are correctly annotated by a keyword and all the images annotated by the model using that keyword [40].

$$P_e = \frac{m_3}{m_2} \tag{6}$$

In the context of model annotation, precision describes the relationship between the images with a given keyword annotated correctly and all the images that have that keyword represented by the model [41].

$$R_e = \frac{m_3}{m_1} \tag{7}$$

To get an overview of the performance of an annotation system, we calculate the average accuracy and reminders across the entire V-size vocabulary [40]:

$$P = \frac{1}{V}\sum_{e=1}^{V} P_e \tag{8}$$

$$R = \frac{1}{V}\sum_{e=1}^{V} R_e \tag{9}$$

6) SCORE E

E-score combines the two reminder and precision measurements into a synthetic quality measurement that can be compared easily [41]:

$$E = 1 - \frac{2}{\frac{1}{P} + \frac{1}{R}} \tag{10}$$

7) F-MEASURE

F-measures are harmonic averages weighted between recall and accuracy [52]:

$$F_\alpha = \frac{\left(1 + \alpha^2\right)(PR)}{a^2 P + R} \tag{11}$$

where $\alpha >= 0$.

TABLE 3. Shows experimental results from our three datasets to demonstrate the competitive.

Dataset	Corel5k				ESPGame				Laprtc12			
Measure	P	R	F	N+	P	R	F	N+	P	R	F	N+
SEM[2]	0.37	0.52	0.43		0.38	0.42	0.4	258	0.41	0.39	0.4	284
2PKNN[4]	0.31	0.43	0.49	190	0.40	0.39	0.49	255	0.37	0.41	0.46	279
FastTag[3]	0.32	0.39	0.36	166	0.39	0.35	0.29	247	0.32	0.34	0.33	280
TagProp[43]	0.33	0.42	0.37		0.37	0.27	0.32		0.46	0.35	0.4	
JEC[44]	0.27	0.27	0.29		0.22	0.25	0.23		0.28	0.29	0.28	
three-pass KNN[55]	0.39	0.55	0.43	198	0.37	0.35	40.8	259	0.51	0.37	0.37	278
PLSA-WORDS[56]	0.20	0.30	0.23	129	0.20	0.24	0.21	201	0.23	0.25	0.23	207
GAN[57]	0.38	0.47	0.41	197	-	-	-	-	0.44	0.38	0.43	199
convolutional features [58]	0.26	0.41	0.32	161	0.37	0.33	0.37	258	0.46	0.32	0.38	258
Proposed Model	0.40	0.55	0.42	200	0.38	0.46	0.50	260	0.42	0.41	0.39	280

The parameter α allows us to assign more or less weight to accuracy. When $\alpha = 1$, recall and accuracy have the same weight. In this case, measure F can be represented using the E score as shown in the following equation [53]:

$$E = 1 - \frac{2}{\frac{1}{P} + \frac{1}{R}} \tag{12}$$

8) N+

Measure N+ is another measure used in annotated systems, the number of words correctly assigned to at least one test image (i.e., the number of words with strictly positive reminders). N+ is a measure of the number of words used in an annotation. This represents the amount of vocabulary covered by the method [42].

IX. EVALUATION CRITERIA SELECTED

The performance of the newly proposed improved AIA scheme was assessed through the design and implementation using the standard dataset. In this paper, we have chosen, like the majority of state-of-the-art works, measurements by word. The performance of the designed AIA was evaluated in terms of various measures: recall, accuracy, F measure and N+. we have used the annotation rate, which is part of the annotation measures [51].

X. EXPERIMENTS RESULTS

The proposed code has been implemented in the Keras (Chollet, 2015), a public deep learning software, based on Tensorflow [6]. Keras was used to initialize the weights in neural networks. All layers in the deep network were initialized simultaneously with the ADADELTA [43]. The complete network was trained using the Dell Precision T1700 CPU system with a 16GB memory. We assessed the computing classification accuracy of a deep learning system using the procedure described in the section 3. The summary of the proposed CNN configuration using the combination between Y, Cb and Cr color channel of image based on Kears library gives the Figure as follows: Table 3 shows the average precision, recall, and F-measure for the CNN model. for each dataset A comparison of experimental data is shown in Fig 18. Table 3 and Figure 18 illustrate how our method has improved with other methods 2PKNN, SEM, and GAN, that consider are more applicable to the annotation task due to their improved precision and recall. In addition, we achieve higher recall and F-values when we process espGame and laprtc12. This paper's primary objective was to propose and implement a new AIA system based on automatic feature extraction and object learning representation that would select the best features. Our model has the highest F-value out of all of them, which indicates its effectiveness.

Table 4 illustrates the annotation of two examples from both the training and testing datasets. When the original images with fewer labels in Table 4 are used for the training set, our proposed method extends the labels effectively. An image can also retain its original labels. By using the method for the test subset, each dataset is effectively annotated. The experimental results were analyzed for Corel5k, ESP Game, and the IAPR TC-12 datasets, respectively.

1) Proposed approach's annotation performance in Corel5k is as follows:

Results of P: CNN-SLT approach provides the highest P, which is 0.40, and GAN reached P to 0.38.

Results of R: CNN-SLT approach provides the highest R, its value is 0.55.

Results of F1: CNN-SLT approach does not provide the highest F1, which is 0.42, the highest F1 in all compared algorithms is 0.499, and the difference between them is 0.08.

Results of N+: CNN-SLT approach provides the highest N+ has a value of 200, which is greater than the highest 198 in the other five analyzed algorithms, and it improves by at least 3. The annotation performances of CNN-SLT and three-pass KNN are the best AIA techniques, as can be shown from the above comparison.

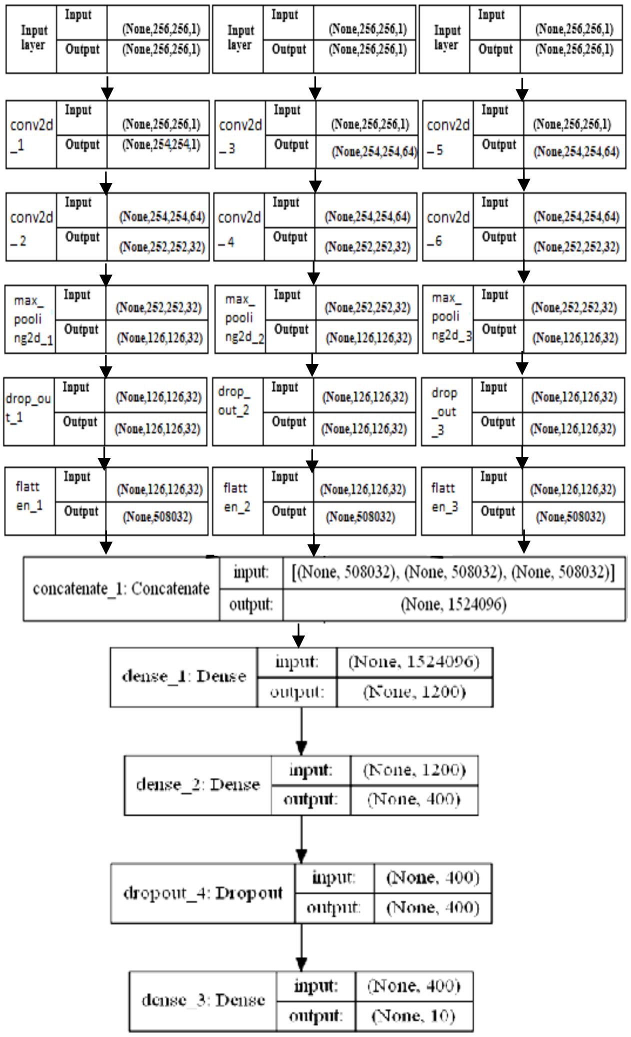

Input layer | Input (None,256,256,1) | Output (None,256,256,1)
conv2d_1 | Input (None,256,256,1) | Output (None,254,254,1)
conv2d_3 | Input (None,256,256,1) | Output (None,254,254,64)
conv2d_5 | Input (None,256,256,1) | Output (None,254,254,64)
conv2d_2 | Input (None,254,254,64) | Output (None,252,252,32)
conv2d_4 | Input (None,254,254,64) | Output (None,252,252,32)
conv2d_6 | Input (None,254,254,64) | Output (None,252,252,32)
max_pooling2d_1 | Input (None,252,252,32) | Output (None,126,126,32)
max_pooling2d_2 | Input (None,252,252,32) | Output (None,126,126,32)
max_pooling2d_3 | Input (None,252,252,32) | Output (None,126,126,32)
drop_out_1 | Input (None,126,126,32) | Output (None,126,126,32)
drop_out_2 | Input (None,126,126,32) | Output (None,126,126,32)
drop_out_3 | Input (None,126,126,32) | Output (None,126,126,32)
flatten_1 | Input (None,126,126,32) | Output (None,508032)
flatten_2 | Input (None,126,126,32) | Output (None,508032)
flatten_3 | Input (None,126,126,32) | Output (None,508032)
concatenate_1: Concatenate | input: [(None, 508032), (None, 508032), (None, 508032)] | output: (None, 1524096)
dense_1: Dense | input: (None, 1524096) | output: (None, 1200)
dense_2: Dense | input: (None, 1200) | output: (None, 400)
dropout_4: Dropout | input: (None, 400) | output: (None, 400)
dense_3: Dense | input: (None, 400) | output: (None, 10)

FIGURE 17. The proposed CNN model configuration using the Y, Cb and Cr color channel as input features representation of image.

2) Our approach's annotation performance in ESP Game is as follows:

Results of P: CNN-SLT approach and SEM provide the highest P, which is 0.38.

Results of R: CNN-SLT approach provides the highest R than other algorithms and its value is 0.46, and SEM reached R to 0.42.

F1 results show that the CNN-SLT technique has the highest F1, 0.50, while the highest F1 in all examined algorithms is 0.49, with a difference of 0.1.

The highest N+ is provided by the CNN-SLT method with three-pass KNN, with a value of 260, which is greater than the highest 259 in the other nine analyzed algorithms, and an improvement of at least 3 over the compared algorithms.

Table 1 shows that, while CNN-SLT approach and SEM provide the largest N+, the difference in F1 between CNN-SLT approach and 2PKNN is 0.1, indicating that CNN-SLT approach outperforms 2PKNN in terms of annotation performance.

TABLE 4. CNN-SLT examples for each dataset.

Corel-5k			
Manual labeling	beach, people, rocks, Kauai	building, bus, tree	elephant, river, tree
prediction	beach, person, rocks, Kauai, water, Clouds	bus, building, tree, sky, house, land	elephant, tree, land, river, sky
ESP-GAME			
Manual labeling	Forest, mountain, river, sky, tree, water	stone, white, cow, dirt, tail, bull, grass,	person, sky, river, lawn, road
prediction	Forest, mountain, river, sky, tree, water, land	stone, white, cow, dirt, tail, bull, grass, land, wall	grass, sky, river, lawn, road, girl, slut
IAPRT C-12			
Manual labeling	city, house, roof, sky, valley, view	Fountain, tree	grandstand, lawn, player, roof, round, stadium
prediction	city, house, roof, sky, valley, view, tree, Clouds	Tree, sky, building, cloud, lawn, land	grandstand, lawn, player, roof, round, stadium, lights, people

3) Our approach's annotation performance in IAPR TC-12 is as follows:

P results show that convolutional features the technique has the highest P (0.46), while CNN-SLT has value P (0.42), the lowest P (0.41) is PLSA-WORDS.

The R value for the CNN-SLT and 2PKNN technique is 0.41, which is higher than the R value for the other algorithms.

F1 results: The highest F1 is 0.39 in the CNN-SLT method and 2PKNN, while the highest F1 in all other algorithms is 0.46. The highest N+ is provided by the CNN-SLT technique, with a value of 280, which is equivalent to FastTag 280 in the other nine analyzed algorithms, and an improvement of at least one over the compared algorithms.

We also find that, although CNN-SLT approach have provide the highest P, R than other algorithms and our approach with 2PKNN have provide the highest F1, the difference in

TABLE 5. Shows experimental results from other three datasets MIML, MSRC and Laprtc12.

Dataset	MIML				MSRC				Laprtc12			
Measure	P	R	F	N+	P	R	F	N+	P	R	F	N+
CNN-AT[48]	0.73	0.749	0.740		0.718	0.832	0.771		0.41	0.39	0.4	
CNN-ECC[49]	0.73	0.758	0.747		0.706	0.747	0.726		0.46	0.35	0.4	
CNN-THOP[50]	0.80	0.776	0.787		0.761	0.783	0.771		0.28	0.29	0.28	
Proposed Model	0.81	0.778	0.82	290	0.776	0.778	0.79	280	0.42	0.44	0.39	286

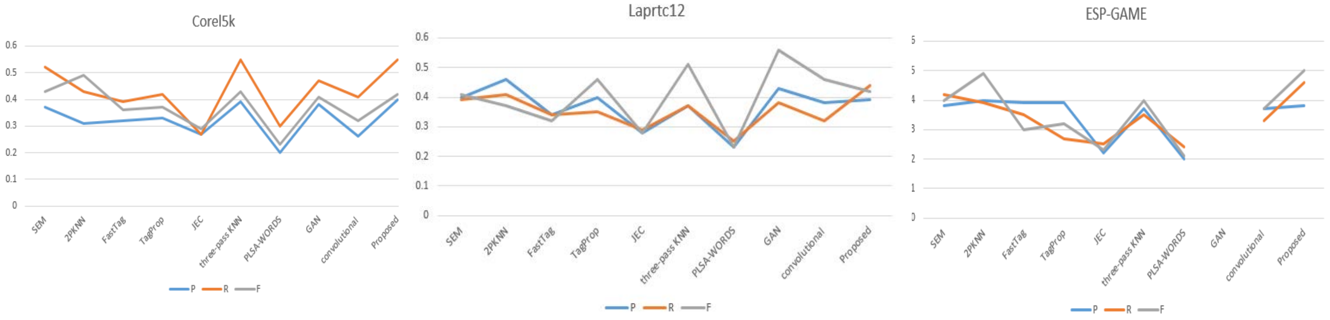


FIGURE 18. A comparison of the CNN-SLT model to others on three datasets.

N+ between CNN-SLT approach and 2PKNN is 7, indicating that CNN-SLT approach has superior annotation performance than 2PKNN and other algorithms. Furthermore, despite the fact that the difference in P between CNN-SLT approach and GAN is only 0.02 and R is only 0.08, the difference in N+ between CNN-SLT approach and GAN is only 3, the difference in F1 between CNN-SLT approach and GAN is 0.1, which is quite substantial. This shows that the CNN-SLT technique outperforms GAN in terms of annotation performance. Furthermore, we compare the annotation performance of CNN-SLT with 2PKNN, and we find that the annotation performance of CNN-SLT is always better than 2PKNN, indicating that CNN-SLT may optimize the annotating result.

XI. COMPARISON WITH OTHER CNN METHODS

This study uses combines and merges the features extracted with the CNN architecture. Our study investigates how different CNN architectures affect experiment results based on three datasets. In Table 5 you will find a summary of the results. Observed in Table 5 more detailed network architectures improve experimental results on datasets of appropriate size. In order to construct the network architecture, we use multi features, to extract shape using Slantlet Transform, second to extract color using YCbCr Colour Space and to extract texture features using Local Binary Pattern (LBP) [44]. Our proposed is compared with traditional methods, comprising shown in table 5, Additionally, deep learning techniques such as deep convolutional neural network (CNN) and k-nearest neighbor's algorithm (KNN) [45] have become increasingly popular. In Table 5, you can see the results for three different datasets of the experiment. Table 5 shows that CNNs perform noticeably better in terms of the investigated indexes than traditional machine learning methods for multilabel annotation.

In the Laprtc12 dataset for natural scenes, average precision is improved in comparison with other methods. Compared to the Laprtc12 dataset, there are improvements in recall [47].

XII. CONCLUSION

This paper has presented an annotation that uses CNN features and neighbors to represent each image by using the CNN feature with Slantlet Transform. Furthermore, an algorithm for semantic extension is presented along with detailed implementations. The researchers in this study assembled all the information from an image using low-level and high-level features, such as shape, texture and color. Automatic features extraction will be discussed to The distributed representation approach included the encoding and storage of information regarding the image features.

Our study investigates how different CNN architectures affect experiment results based on three datasets. Lastly, the effectiveness of this model for image annotation was demonstrated through experimental results from three data sets. The experimental results for three public datasets—COREL5K, ESP-Game, and Iaprtc12—indicate that the average precision of the CNN-SLT is 0.40, 0.38 and 0.42%, respectively, and the average recall is 0.55, 0.46 and 0.41, respectively. The F1 value reaches 0.42, 0.50 and 0.39, respectively, and the N+ also reaches 200, 260 and 280, respectively. While some of the research on image annotation systems tends towards high accuracy and squat recall, the perfect system can be achieved by balancing precision and recall while maintaining precision. The proposed method achieves a balance between precision and recall using SLT and selects the most appropriate features. Meanwhile, As the training phase of some image annotation models requires considerable computation time and complexity, they become computationally intensive when large training datasets are used. This method has been successful in terms of computational efficiency, which is what researchers have always struggled with.

Declaration: all authors contributed equally scientifically and have no conflict to declare for this research

ACKNOWLEDGMENT

The authors would like to acknowledge the support of Prince Sultan University for paying the Article Processing Charges (APC) of this publication.

REFERENCES

[1] M. Yasmin, S. Mohsin, and M. Sharif, "Intelligent image retrieval techniques: A survey," *J. Appl. Res. Technol. Universidad Nacional Autonoma de Mexico, Centro de Ciencias Aplicadas Y Desarrollo Tecnologico*, vol. 12, no. 1, pp. 87–103, 2014, doi: 10.1016/S1665-6423(14)71609-8.

[2] Z. Lu, P. Han, L. Wang, and J. R. Wen, "Semantic sparse recoding of visual content for image applications," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 176–188, Jan. 2015, doi: 10.1109/TIP.2014.2375641.

[3] Y. Mistry, D. T. Ingole, and M. D. Ingole, "Content based image retrieval using hybrid features and various distance metric," *J. Electr. Syst. Inf. Technol.*, vol. 5, no. 3, pp. 874–888, Dec. 2018, doi: 10.1016/j.jesit.2016.12.009.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, 2013, pp. 3111–3119.

[5] Y. Ma, Y. Liu, Q. Xie, and L. Li, "CNN-feature based automatic image annotation method," *Multimedia Tools Appl.*, vol. 78, no. 3, pp. 3767–3780, 2019.

[6] M. Chen, A. Zheng, and K. Weinberger, "Fast image tagging," in *Proc. Int. Conf. Int.*, 2013, pp. 1274–1282.

[7] Y. Verma and C. V. Jawahar, "Image annotation using metric learning in semantic neighbourhoods," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 836–849, doi: 10.1007/978-3-642-33712-3_60.

[8] H. Fu, Q. Zhang, and G. Qiu, "Random forest for image annotation," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 86–99.

[9] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "CNN-RNN: A unified framework for multi-label image classification," in *Proc. IEEE Conf. Computer Vis. Pattern Recognit.*, Jun. 2016, pp. 2285–2294.

[10] V. N. Murthy, A. Sharma, V. Chari, and R. Manmatha, "Image annotation using multi-scale hypergraph heat diffusion framework," in *Proc. ACM Int. Conf. Multimedia Retr.*, Jun. 2016, pp. 299–303, doi: 10.1145/2911996.2912055.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[12] E. Park, X. Han, T. L. Berg, and A. C. Berg, "Combining multiple sources of knowledge in deep CNNs for action recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–8, doi: 10.1109/WACV.2016.7477589.

[13] M. Egmont-Petersen, R. D. de Ridder, and H. Handels, "Image processing with neural networks—A review," *Pattern Recognit.*, vol. 35, no. 10, pp. 2279–2301, 2002, doi: 10.1016/S0031-3203(01)00178-9.

[14] J. Dolz, C. Desrosiers, and I. A. Ben, "IVD-Net: Intervertebral disc localization and segmentation in MRI with a multi-modal UNet," in *Proc. MICCAI IVD Challenge, (IVD)*, 2018, pp. 1–7, doi: 10.13140/RG.2.2.23756.4672.

[15] M. Atzori, M. Cognolato, and H. Müller, "Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands," *Frontiers Neurorobot.*, vol. 10, pp. 1–10, Sep. 2016, doi: 10.3389/fnbot.2016.00009.

[16] H. Abubakar, A. Muhammad, and S. Bello, "Ants colony optimization algorithm in the Hopfield neural network for agricultural soil fertility reverse analysis," *Iraqi J. Comput. Sci. Math.*, pp. 32–42, Jan. 2022.

[17] H. Saggion and T. Poibeau, "Automatic text summarization: Past, present and future," in *Multi-Source, Multilingual Information Extraction and Summarization*. Berlin, Germany: Springer, 2013, pp. 3–21.

[18] T. Zeng and S. Ji, "Deep convolutional neural networks for multi-instance multi-task learning," in *Proc. IEEE Int. Conf. Data Mining*, Nov. 2015, pp. 579–588, doi: 10.1109/ICDM.2015.92.

[19] Q. Liao, L. Jiang, X. Wang, C. Zhang, and Y. Ding, "Cancer classification with multi-task deep learning," in *Proc. Int. Conf. Secur., Pattern Anal., Cybern. (SPAC)*, Dec. 2017, pp. 76–81, doi: 10.1109/SPAC.2017.8304254.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1–9, doi: 10.1016/j.protcy.2014.09.007.

[21] A. Ferreira and G. Giraldi, "Convolutional neural network approaches to granite tiles classification," *Expert Syst. Appl.*, vol. 84, pp. 1–11, Oct. 2017, doi: 10.1016/j.eswa.2017.04.053.

[22] Y. Liu and X. An, "A classification model for the prostate cancer based on deep learning," in *Proc. 10th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Oct. 2017, pp. 1–6, doi: 10.1109/CISP-BMEI.2017.8302240.

[23] P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer, "Real-time classification and sensor fusion with a spiking deep belief network," *Frontiers Neurosci.*, vol. 7, pp. 1–13, Oct. 2013, doi: 10.3389/fnins.2013.00178.

[24] F. M. Rammo and N. M. Al-Hamdani, "Detecting the speaker language using CNN deep learning algorithm," *Iraqi J. Comput. Sci. Math.*, vol. 3, no. 1, pp. 43–51, 2022.

[25] G. Quellec, M. Lamard, G. Cazuguel, B. Cochener, and C. Roux, "Adaptive nonseparable wavelet transform via lifting and its application to content-based image retrieval," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 25–35, Jan. 2010, doi: 10.1109/TIP.2009.2030479.

[26] N. N. Ghuge and B. D. Patil, "Multi resolution features of content based image retrieval," *ACEEE Int. J. Signal Image Process.*, vol. 4, no. 3, pp. 65–71, 2013.

[27] I. W. Selesnick, "The slantlet transform," *IEEE Trans. Signal Process.*, vol. 47, no. 5, pp. 1304–1313, May 1999, doi: 10.1109/78.757218.

[28] G. Muhammad, "Copy move image forgery detection method using steerable pyramid transform and texture descriptor," in *Proc. IEEE EuroCon*, Jul. 2013, pp. 1586–1592, doi: 10.1109/EUROCON.2013.6625188.

[29] G. Zhang, "Boosting local binary pattern (LBP)-based face recognition," in *Proc. SINOBIOMETRICS Adv. Biometric Person Authentication*, 2004, pp. 179–186, doi: 10.1007/978-3-540-30548-4_21.

[30] P. Duygulu, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *Proc. Eur. Conf. Comput. Vis.*, 2002, pp. 97–112, doi: 10.1007/3-540-47979-1_7.

[31] A. L. Von and L. Dabbish. (2004). *Labeling Images with a Computer Game*. Accessed: Oct. 12, 2019. [Online]. Available: http://www.espgame.org

[32] H. J. Escalante, C. A. Hernández, J. A. Gonzalez, A. López-López, M. Montes, E. F. Morales, L. E. Sucar, L. Villaseñor, and M. Grubinger, "The segmented and annotated IAPR TC-12 benchmark," *Comput. Vis. Image Understand.*, vol. 114, no. 4, pp. 419–428, 2010, doi: 10.1016/j.cviu.2009.03.008.

[33] H. Kwasnicka and M. Paradowski, "On evaluation of image auto-annotation methods," in *Proc. 6th Int. Conf. Intell. Syst. Design Appl.*, Oct. 2006, pp. 353–358, doi: 10.1109/ISDA.2006.253861.

[34] Y. A. Aslandogan, C. Thier, C. T. Yu, J. Zou, and N. Rishe, "Using semantic contents and WordNet in image retrieval," in *Proc. 20th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 1997, pp. 286–295. Accessed: Oct. 12, 2019. [Online]. Available: http://cake.fiu.edu/Publications/Aslandogan+al-97-U.S..Using_semantic_contents_and_Wordnet_in_image_retrieval.published-scanned.pdf

[35] K. Barnard, "Matching words and pictures," *J. Mach. Learn. Res.*, vol. 3, pp. 1107–1135, Mar. 2003. Accessed: Oct. 12, 2019. [Online]. Available: http://www.jmlr.org/papers/v3/barnard03a.html

[36] S. Barrat and S. Tabbone, "Modeling, classifying and annotating weakly annotated images using Bayesian network," *J. Vis. Commun. Image Represent.*, vol. 21, no. 4, pp. 355–363, May 2010, doi: 10.1016/j.jvcir.2010.02.010.

[37] M. Bianchini and F. Scarselli, "On the complexity of neural network classifiers: A comparison between shallow and deep architectures," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 8, pp. 1553–1565, Aug. 2014, doi: 10.1109/TNNLS.2013.2293637.

[38] L. Breiman, "Documentation for R package randomForest," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[39] X. Cao, H. Zhang, X. Guo, S. Liu, and D. Meng, "SLED: Semantic label embedding dictionary representation for multilabel image annotation," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2746–2759, Sep. 2015, doi: 10.1109/TIP.2015.2428055.

[40] Y. Cao, X. Liu, J. Bing, and L. Song, "Using neural network to combine measures of word semantic similarity for image annotation," in *Proc. IEEE Int. Conf. Inf. Autom.*, Jun. 2011, pp. 833–837, doi: 10.1109/ICINFA.2011.5949110.

[41] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, "Supervised learning of semantic classes for image annotation and retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 394–410, Mar. 2007, doi: 10.1109/TPAMI.2007.61.

[42] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*.

[43] O. Chapelle, "Support vector machines for histogram-based image classification," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1055–1064, Sep. 1999. Accessed: Oct. 12, 2019. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.459.9821&rep=rep1&type=pdf

[44] M. Guillaumin *et al.*, "TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Kyoto, Japan, 2010.

[45] A. Makadia, V. Pavlovic, and S. Kumar, "A new baseline for image annotation," *Int. J. Comput. Vis.*, vol. 90, pp. 88–105, Oct. 2010.

[46] M. M. Adnan, M. S. M. Rahim, A. Rehman, Z. Mehmood, T. Saba, and R. A. Naqvi, "Automatic image annotation based on deep learning models: A systematic review and future challenges," *IEEE Access*, vol. 9, pp. 50253–50264, 2021.

[47] Y. Yuan, G. Xun, Q. Suo, K. Jia, and A. Zhang, "Wave2 Vec: Deep representation learning for clinical temporal data," *Neurocomputing*, vol. 324, pp. 31–42, Jan. 2019, doi: 10.1016/j.neucom.2018.03.074.

[48] F. Wu, Z. Wang, Z. Zhang, Y. Yang, J. Luo, W. Zhu, and Y. Zhuang, "Weakly semi-supervised deep learning for multi-label image annotation," *IEEE Trans. Big Data*, vol. 1, no. 3, pp. 109–122, Sep. 2015.

[49] W. Zhang, H. Hu, H. Hu, and J. Yu, "Automatic image annotation via category labels," *Multimedia Tools Appl.*, vol. 79, nos. 17–18, pp. 11421–11435, May 2020.

[50] J. Cao, A. Zhao, and Z. Zhang, "Automatic image annotation method based on a convolutional neural network with threshold optimization," *PLoS ONE*, vol. 15, no. 9, Sep. 2020, Art. no. e0238956.

[51] M. M. Adnan, M. S. M. Rahim, K. Al-Jawaheri, and K. Neamah, "A review of methods for the image automatic annotation," *J. Phys., Conf. Ser.*, vol. 1892, no. 1, Apr. 2021, Art. no. 0120021.

[52] M. H. Ali, K. Al-Jawaheri, M. M. Adnan, A. Aasi, and A. H. Radie, "Improved intrusion detection accuracy based on optimization fast learning network model," in *Proc. 3rd Int. Conf. Eng. Technol. Appl. (IICETA)*, Sep. 2020, pp. 198–202.

[53] K. A. Kadhim, F. Mohamed, Z. N. Khudhair, and M. H. Alkawaz, "Classification and predictive diagnosis earlier Alzheimer's disease using MRI brain images," in *Proc. IEEE Conf. Big Data Anal. (ICBDA)*, Nov. 2020, pp. 45–50.

[54] K. A. Kadhim, F. Mohamed, and Z. N. Khudhair, "Deep learning: Classification and automated detection earlier of Alzheimer's disease using brain MRI images," *J. Phys., Conf. Ser.*, vol. 1892, no. 1, Apr. 2021, Art. no. 012009.

[55] H. Li, W. Li, H. Zhang, X. He, M. Zheng, and H. Song, "Automatic image annotation by sequentially learning from multi-level semantic neighborhoods," *IEEE Access*, vol. 9, pp. 135742–135754, 2021.

[56] T. Uricchio, L. Ballan, L. Seidenari, and A. Del Bimbo, "Automatic image annotation via label transfer in the semantic space," *Pattern Recognit.*, vol. 71, pp. 144–157, Nov. 2017.

[57] J. Liu and W. Wu, "Automatic image annotation using improved Wasserstein generative adversarial networks," *IAENG Int. J. Comput. Sci.*, vol. 48, no. 3, pp. 1–7, 2021.

[58] Y. Chen, L. Liu, J. Tao, X. Chen, R. Xia, Q. Zhang, J. Xiong, K. Yang, and J. Xie, "The image annotation algorithm using convolutional features from intermediate layer of deep learning," *Multimedia Tools Appl.*, vol. 80, no. 3, pp. 4237–4261, Jan. 2021.