Computers, Materials & Continua
DOI: 10.32604/cmc.2022.024975
Article

Tech Science Press

Historical Arabic Images Classification and Retrieval Using Siamese Deep Learning Model

Manal M. Khayyat[1,2], Lamiaa A. Elrefaei[2,3] and Mashael M. Khayyat[4,*]

[1]Computer Science Department, Umm Al-Qura University, Makkah, Saudi Arabia
[2]Computer Science Department, King Abdulaziz University, Jeddah, Saudi Arabia
[3]Electrical Engineering Department, Faculty of Engineering at Shoubra, Benha University, Cairo, Egypt
[4]Department of Information Systems and Technology, College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia
*Corresponding Author: Mashael M. Khayyat. Email: mkhayyat@uj.edu.sa
Received: 06 November 2021; Accepted: 11 January 2022

Abstract: Classifying the visual features in images to retrieve a specific image is a significant problem within the computer vision field especially when dealing with historical faded colored images. Thus, there were lots of efforts trying to automate the classification operation and retrieve similar images accurately. To reach this goal, we developed a VGG19 deep convolutional neural network to extract the visual features from the images automatically. Then, the distances among the extracted features vectors are measured and a similarity score is generated using a Siamese deep neural network. The Siamese model built and trained at first from scratch but, it didn't generated high evaluation metrices. Thus, we re-built it from VGG19 pre-trained deep learning model to generate higher evaluation metrices. Afterward, three different distance metrics combined with the Sigmoid activation function are experimented looking for the most accurate method for measuring the similarities among the retrieved images. Reaching that the highest evaluation parameters generated using the Cosine distance metric. Moreover, the Graphics Processing Unit (GPU) utilized to run the code instead of running it on the Central Processing Unit (CPU). This step optimized the execution further since it expedited both the training and the retrieval time efficiently. After extensive experimentation, we reached satisfactory solution recording 0.98 and 0.99 F-score for the classification and for the retrieval, respectively.

Keywords: Visual features vectors; deep learning models; distance methods; similar image retrieval

1 Introduction

The accurate image classification and retrieval concerned many researchers. However, most of the efforts handled clear modern images, which made the search and retrieval process much easier. The

authors reached that the deep neural networks work well in extracting the features from the images automatically and therefore can retrieve similar images perfectly.

Deep neural networks usually involve mathematical algorithms that enable them to solve complex problems effectively. They contain a number of hidden layers that enable them to build knowledge on top of each other [1]. Hence, as much as the neural network is deeper, as much as, the network builds a better understanding of the problem to be solved. Besides the number of layers, there are the learning parameters, which affect the learning process. Therefore, the developers need to augment the parameters and balance their dataset to generate higher performance results.

One of the most effective approaches used in training deep neural networks is the transfer learning approach. This approach expedites the learning process. Because it uses the previously learned knowledge from specific huge dataset images on a new unseen dataset images.

In this study, we challenged ourselves by considering the historical low-quality text-based images. We developed a deep neural network to measure the similarities among the extracted visual features and retrieve the most similar images to a historical query image. Our contributions are summarized as follows:

1. Utilize both "one-shot learning" strategy along with the "weighted cross-entropy" algorithm to assist in balancing the dataset and, therefore, classify the images effectively.
2. Classify the images using a pre-trained deep learning model and enter the classified features vectors into a Siamese model that is built according to the architecture of the VGG19 deep learning model to minimize the distances among the classified images further.
3. Compute the similarity scores using three different distance metrics used in conjunction with the Siamese Sigmoid activation function, looking for the metric that produces the highest similarity results.
4. Optimize the retrieval of the top-k similar images to a user query image without the need for any images' segmentation pre-process through utilizing the GPU.

Rest of the paper is organized as follows: Section 2 reviews related work on the field. Section 3 discusses the proposed method, including its mathematical representation and development, as well as, it explains the employed strategies for the learning and for image retrieval. Section 4 conducts the experiments and clarifies their results and, Section 5 concludes the paper.

2 Literature Review

Most recent studies utilized the deep convolutional neural networks for image classification and retrieval and proved their success. Mehmood et al. [2] utilized the VGG-16 deep convolutional neural network to automatically extract the visual features from Alzheimer's disease images. Then, they used the Siamese deep neural network to accurately find similar images and classify them. To overcome the small number of training images, the authors did an augmentation step. Finally, they were able to reach 99.05% accurate classification of Alzheimer's disease images.

Tian et al. [3] experimented retrieving similar images using a Siamese model but, with an enhanced loss function that is different than the commonly used logistics and triplet loss functions. They started their work by training samples from "ILSVRC2015" then, evaluate their model using the images of "OTB2015" dataset. The enhanced loss function linked all of the sample images using a dense layer. The authors experimented their model with different parameters reaching 60% Area Under Curve (AUC).

An-Bing et al. [4] developed a deep Siamese near-infrared spectroscopy model to predict drugs. They begin by collecting the data set images and label the similar pairs with "0" and the un-similar pairs with "1". Afterward, they trained the developed deep Siamese model using CUDA GPU by running the model 60 epochs. The authors recorded above 97% accurate prediction.

Baraldi et al. [5] developed a Siamese deep learning model to discover similar videos. They began their work by splitting the videos into pictures then, develop a Convolutional Neural Network (CNN) that is similar in its architecture and layers to the Caffe neural network. The authors trained their network on the images of two popular datasets, which are ImageNet and Places datasets. Afterward, they tuned the developed CNN and extracted the visual features from their videos' images. The ReLU activation function was used to measure the similarities among the extracted visual feature vectors. Regarding the textual contents within the videos, they were extracted using the clustering k-means method. Then, the similarities among the extracted words were measures using the Cosine distance method. Finally, the authors merged both the visual and the textual similarities into one score using the Gaussian kernel. The authors evaluated their model and recorded around 64% successful detection of the video scenes.

Li et al. [6] proposed a ResNet50-driven Siamese model that accepts (127 x 127) colored image and then retrieve the most similar images to the query image. The authors started by modifying the original ResNet50 CNN by minimizing the strides at the least blocks to include only eight pixels. They also added additional convolutional layer to reduce the number of accepted channels. Afterward, they trained their model on the TrackingNet dataset images recording 73.3% Area Under Curve (AUC).

Chen et al. [7] developed two different deep Siamese networks to extract the features from the Very High Resolution (VHR) images within the GF dataset. The first developed Siamese model was not fully connected, while the second enhanced model is fully connected. The authors reached that the second model is outperforming the first one recording 97.89% overall accuracy. Similarly, Chaudhuri et al. [8] designed a Siamese model for retriving the VHR remote sensing images. The authors segmented the dataset images into regions. Then, they employed the adjacency graph method to find-out the closest regions to each other and hence, figure-out the similar images. The others evaluated their model using two datasets named: UC-Merced and the PatternNet datasets. They calculated the mean Average Precision (mAP) and recorded 69.89% and 81.79% using the UC-Merced and the PatternNet datasets respectively.

Kande et al. [9] designed a generative Siamese deep learning model to retrieve the Spectral-Domain Optical Coherence Tomography (SDOCT) images. The researchers proposed joining the loss generated from the Siamese model with the restoration loss that is generated from the CNN as this combination resulted in more similar images to the query image. To evaluate their model, the authors computed the Texture Preservation (TP) index and recoded 68%.

Radenovic et al. [10] recommended fun-tuning a deep VGG-CNN on an annotated 3D images. They employed the Siamese learning strategy to extract the visual features from the Oxford5k and Paris6k datasets' images. The authors computed the mean Average Precision to assess their model and reached 91.9%. Similarly, Zhou et al. [11] utilized the Sketch-Based Local Binary Pattern (SBLBP) method to extract the visual features from sketched 3D images. Then, they developed a Siamese deep learning model to measure the similarities among the extracted featutres and retrieve similar images. The authors experimented their model using two popular datasets, which are the National Taiwan University (NTU) and a sketched dataset from Eitz et al. Finally, they recorded 60% and 39% Precision on the NTU and the sketched datasets respectively.

Koch et al. [12] used the one-shot learning strategy to train their developed Siamese neural network on the Omniglot dataset. The dataset contains images of handwritten alphabets. The authors experimented three different dataset sizes for training their Siamese network and they concluded that, as much as, the number of the training samples increases, as much as, the verification accuracy also increases reaching 93.42% using 150,000 training images.

Ong et al. [13] developed VGG16 CNN to extract the visual features from the images. Afterward, they classified the images using the Fisher Vector (FV) encoders. The authors employed the Euclidean metric to measure the differences between the images and eventually retrieve similar ones. They were able to record 81.5% mAP using the Oxford dataset and 82.5% mAP on the Paris datasets. On the other hand, Qiu et al. [14] also developed a Siamese model combined with an Euclidean metric to retireve the similar images, but they utilized the ResNet CNN instead of VGG to discover the loop closure. The authors evaluated their model using the TUM dataset and recorded 87.7% precision.

Wiggers et al. [15] prefered utilizing the AlexNet CNN in conjunction with the Siamese neural network. They experimented their model using the public "Tobacco800" dataset images. The authors also tested the effects of different feature vector sizes. They concluded that, as much as, the feature size increase, as much as, the model record better accurate results reaching 94.4% mAP.

From the studied literature, we reach that a number of researchers employed the Saiemse model and proved its success in measuring the similarities among the images. However, none of the studies tested the efficiency of the model on the Arabic handwritten characters. Also, none of the studies experimented with different distance metrics in conjunction with the Siamese model. Thus, we aim in this study to experiment more than one distance metric with the Siamese model and also execute the model using both the CPU and the GPU to optimize the image retrieval task.

3 Methodology

This section begins by explaining the development of the proposed method, including its mathematical representation and learning strategy. Afterward, it explains the method of matching the images and retrieving similar images to a user query image.

3.1 Proposed Method

The proposed method consists of two main steps, which are: the image classification and the similarity measurement. The image classification step begins by entering all the dataset images into a pre-trained VGG19 deep convolutional neural network to automatically extract the global high-level features from the images and classify them as explained in [16]. The feature vector of the user query image denoted as FV(a) and the feature vector of all other images stored in the dataset denoted as FV(b), are saved in a database to be able to access them easier and faster during the second step.

The second step, which is the similarity measurement, takes the outputs from the images' classification step, which are the feature vector of the user query image and the features vectors of all classified images stored in the dataset and enters them as inputs to the Siamese model as illustrated in Fig. 1.

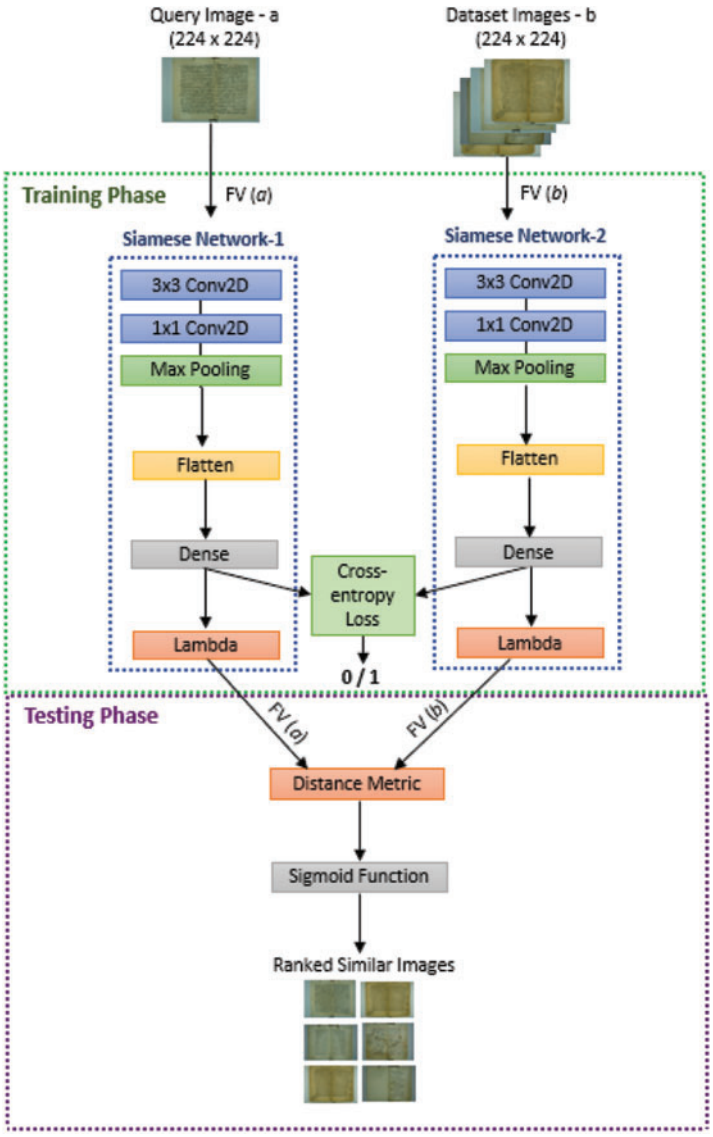

Figure 1: Training and testing the Siamese model for retrieval

From Fig. 1, we notice that during the training phase, the Siamese model enters each one of the input images features vectors into one of the twin Siamese networks. The model uses a max-pooling layer to reduce the spatial dimensionality that exists within the tensors, followed by a flatten layer to convert the two-dimensional matrix into one vector. Then, a dense layer including the "Sigmoid" activation function, to predict the similarities among images. Hence, the last employed layer by the model during the training is the "Cross-entropy loss" shared layer, which includes only one neuron to classify the images as either similar denoted by (1) or not similar, denoted by (0).

In the testing phase, the Siamese model re-produces the feature vector of the images after minimizing the distances among their classified classes and computes the distances between the feature vector of the user query image and the features vectors of all other images in the dataset to generate

the ranked similarity scores after computing the sigmoid function of the computed distances. The final outputs from the model are the ranked top-k similar images to a user query image.

3.1.1 Mathematical Representation of the Siamese Model

The Siamese model doesn't learn any classification; instead it learns to compute the similarity between classified images [17]. It consists of twin convolutional networks that share the same weights and structures and are joint through their calculated weights and the computed loss function by the last fully connected dense layer. The main goal of the loss function is to reduce the difference among the probability distribution of the true labels and the predicted labels. The minimization function presented in Eq. (1) by Lamba [17], ensures that the classified images from the same class preserve analogous feature vector.

$$Ls_i = -\sum_{k=1}^{C} t_k(y_i)\ log\ P(y_i = k|b_i; w_k) \tag{1}$$

where Ls is the loss function of the i^{th} feature in a dataset including a total number of N samples, $i = 1, 2, 3, \ldots, N$.

k is the class and C is the classes' classifier. y_i is the true label in the Y set of true labels, $Y = \{y_i\}_{i=1}^{N}$, $y_i \in \{1, 2, 3, \ldots., C\}$ and $t_k(y_i)$ is the distribution of y_i.

$P(y_i = k|b_i; w_k)$ is the probability distribution of the predicted label. b_i is the binary representation of the i^{th} feature, $B = \{b_i\}_{i=1}^{N}$, $b_i \in \{-1,\ 1\}$. w_k is the weight of the classifier.

After computing the loss function, the Siamese model usually contains a pure static distance metric to compute the exact similarity score of the images. Hence, it subtracts the feature vector of the user query image from the features vectors of all other images stored in the dataset. And after calculating the difference between the two extracted features vectors, it converts the computed difference into one single number using the "Sigmoid" activation function. The output number from the final computed "Sigmoid" function presented in [18] is called the similarity score (SC).

$$SC = \sigma\left(\sum_{j} \propto_j |h^{(j)}_{1,\,L-1} - h^{(j)}_{2,\,L-1}\right) \tag{2}$$

where σ is the "Sigmoid" activation function, j refers to the number of neurons on a layer.

$\propto_j$ is the final computed difference between the two feature vectors by the twins' Siamese neural networks, L represents the number of layers in each Siamese neural network. Hence, $L-1$ is the layer before the last fully connected layer.

$h^{(j)}_{1,\,L-1}$ represents the hidden features falling in the layer before the last fully connected layer within the first Siamese neural network. Similarly, $h^{(j)}_{2,\,L-1}$ represents the hidden features falling in the layer before the last fully connected layer within the second twin of the Siamese neural network.

3.1.2 Siamese Model Development

There are two different developments for the Siamese deep learning model. The first is to develop the model and train from scratch. While the second method is to develop the Siamese model using the structure and the weights of a pre-trained deep learning model. Tab. 1 illustrates the layers, weights, and the overall architecture of the built Siamese model from scratch.

Inspired by Singh [19] the second developed Siamese model is according to the structure of the pre-trained VGG19 deep learning model, as illustrated in Tab. 2.

Table 1: The architecture of Siamese deep neural network

Layer	Type	Output	Parameters	Connected to
Input_1	Input layer	(224, 224, 3)	0	NA
Input_2	Input Layer	(224, 224, 3)	0	NA
Conv2d_1	Conv2D	(148, 98, 64)	1792	Input_1 [0][0]Input_2 [0][0]
Conv2d_2	Conv2D	(148, 98, 64)	4160	Conv2d_1 [2][0]Conv2d_1 [3][0]
Max_pooling2d_1	Maxpooling2D	(74, 49, 64)	0	Conv2d_2 [2][0]Conv2d_2 [3][0]
Conv2d_3	Conv2D	(72, 47, 128)	73856	Max_pooling2d_1 [2][0]Max_pooling2d_1 [3][0]
Conv2d_4	Conv2D	(72, 47, 64)	8256	Conv2d_3 [2][0]Conv2d_3 [3][0]
Max_pooling2d_2	Maxpooling2D	(36, 23, 64)	0	Conv2d_4 [2][0]Conv2d_4 [3][0]
Conv2d_5	Conv2D	(34, 21, 256)	147712	Max_pooling2d_2 [2][0]Max_pooling2d_2 [3][0]
Conv2d_6	Conv2D	(34, 21, 64)	16448	Conv2d_5 [2][0]Conv2d_5 [3][0]
Max_pooling2d_3	Maxpooling2D	(17, 10, 64)	0	Conv2d_6 [2][0]Conv2d_6 [3][0]
Conv2d_7	Conv2D	(15, 8, 256)	147712	Max_pooling2d_3 [2][0]Max_pooling2d_3 [3][0]
Conv2d_8	Conv2D	(15, 8, 64)	16448	Conv2d_7 [2][0]Conv2d_7 [3][0]
Max_pooling2d_4	Maxpooling2D	(7, 4, 64)	0	Conv2d_8 [2][0]Conv2d_8 [3][0]
Flatten_1	Flatten	(1792)	0	Max_pooling2d_4 [2][0]Max_pooling2d_4 [3][0]
Dense_1	Dense	(256)	459008	Flatten_1 [2][0]Flatten_1 [3][0]
Lambda_2	Lambda	(256)	0	Dense_1 [2][0]Dense_1 [3][0]
Dense manuscript	Dense	(1)	257	Lambda_2 [0][0]

From Tabs. 1 and 2 we notice that the Siamese model accepts two input images. In contrast, all other classical deep learning models take only one input image.

Table 2: The architecture of VGG19-Siamese deep learning model

Layer	Type	Output	Parameter	Connected to
Input_1	Input layer	224, 224, 3	0	NA
Input_2	Input layer	224, 224, 3	0	NA
Vgg19	Model	Multiple	20024384	Input_1 [0][0] Input_2 [0][0]
Global_max_pooling 2d_19	Global	512	0	Vgg19 [1][0]
Global_average_pooling 2d_20	Global	512	0	Vgg19 [1][0]
Global_max_pooling 2d_20	Global	512	0	Vgg19 [2][0]
Global_average_pooling 2d_21	Global	512	0	Vgg19 [2][0]
Concatenate_28	Concatenate	1024	0	Global_max_pooling2d_19 [0][0] Global_average_pooling 2d_20 [0][0]
Concatenate_29	Concatenate	1024	0	Global_max_pooling2d_20 [0][0] Global_average_pooling 2d_21 [0][0]
Multiply_29	Multiply	1024	0	Concatenate_28 [0][0] Concatenate_28 [0][0]
Multiply_30	Multiply	1024	0	Concatenate_29 [0][0] Concatenate_29 [0][0]
Subtract _19	Subtract	1024	0	Concatenate_28 [0][0] Concatenate_29 [0][0]
Lambda_10	Lambda	1	0	Concatenate_28 [0][0] Concatenate_29 [0][0]
Subtract _20	Subtract	1024	0	Multiply_29 [0][0] Multiply_30 [0][0]
Multiply_28	Multiply	1024	0	Subtract _19 [0][0] Subtract _19 [0][0]
Concatenate_30	Concatenate	2049	0	Lambda_10 [0][0] Subtract_20 [0][0] Multiply_28 [0][0]
Dense_23	Dense	100	205000	Concatenate_30 [0][0]
Dropout_10	Dropout	100	0	Dense_23 [0][0]
Dense_24	Dense	1	101	Dropout_10 [0][0]

The total number of trainable parameters, within the built Siamese model from scratch, equals 857,649. While it equals 20,229,485 in the Siamese model built from the structure of the VGG19 pre-trained deep learning model.

3.2 Learning Strategy

One-shot learning strategy is employed, which allows the model to predict the true labels after seeing only one example from each class. It is accomplished through defining two customized lists for the training subset, named: images and external. The images customized training list takes one random image from each class and shows it to the model as an example of a similar matching image. In contrast, the external customized training list shows the model one random image that is from any different manuscript than the user query image to show the model an example of non-similar images. This strategy of learning saves training time, and it is useful when some of the dataset classes include only few examples.

Moreover, the "weighted cross-entropy" algorithm used to weight the classes including the minimum number of images more than the classes including a more significant number of images. This algorithm is used to fix the unbalanced images within the dataset. Because the unbalanced ratio of images may result in an accuracy paradox problem were the generated results could be biased and overfitted to the manuscript, including the largest number of images [20].

4 Experiments and Tests Results

In this section, we initially clarify the settings for our experiments then, we begin by experimenting the image classification using the VGG19 deep learning model. Afterward, we experiment the similarity measurement using the developed Siamese model from scratch and compares it with the developed Siamese model using the weights and the structure of the VGG19 pre-trained deep learning model. Moreover, we test three different distance metrics combined with the Siamese model to find the most accurate metric in calculating the similarity scores.

4.1 Settings of the Experiments

The testing machine is "ABS Battelbox" PC, including Ubuntu 16.04 Operating System and Intel Core i7-9700K 3.60 GHz with (8) core processors and Nvidia Gefore RTX 2080. In regard of the used softwares to run the experiments, we utilized Python (3.7.4) programming language with the Jupyter notebook web application interface.

After setting the hardware and software for the experiments, we started by categorizing the dataset used in the study by Khayyat et al. [21], which consists of (8638) historical Arabic manuscripts' images into three main subsets that are the training, the validation, and the testing. 70% of the entire dataset size is allocated for the training purpose. While 30% of the dataset size is divided equally between both the validation and the testing subsets. The model initially trained on the 70% data using ten learning cycles and then tested on the 15% unseen data by the model to evaluate its performance.

4.2 Image Classification

The main goal of the VGG19 deep learning model is to classify the images according to their predicted manuscript id. The recorded evaluation parameters of the classification process are summarized in Tab. 3.

Table 3: Evaluation of the image classification model

Parameter	Value
Accuracy	0.9768
Precision	0.9836
Recall	0.9811
F-score	0.9817

From Tab. 3 we notice that the generated evaluation parameters are all above 97%, which confirms the model's success in classifying the images. Tab. 4 illustrates the calculated precision, recall, and F-score per each classified manuscript.

Table 4: Evaluation parameters per classified manuscript

Manuscript ID	Precision	Recall	F-score	Manuscript ID	Precision	Recall	F-score
1	1.0000	0.8824	0.9375	2	0.7692	0.9524	0.8511
3	1.0000	1.0000	1.0000	4	0.9600	0.8889	0.9231
5	1.0000	1.0000	1.0000	6	1.0000	1.0000	1.0000
7	1.0000	1.0000	1.0000	8	1.0000	1.0000	1.0000
9	1.0000	0.9583	0.9787	10	1.0000	1.0000	1.0000
11	1.0000	1.0000	1.0000	12	0.9545	1.0000	0.9767
13	1.0000	1.0000	1.0000	14	1.0000	1.0000	1.0000
15	1.0000	1.0000	1.0000	16	1.0000	0.9231	0.9600
17	1.0000	1.0000	1.0000	18	1.0000	1.0000	1.0000
19	1.0000	1.0000	1.0000	20	1.0000	1.0000	1.0000
21	1.0000	1.0000	1.0000	22	0.9583	1.0000	0.9787
23	1.0000	1.0000	1.0000	24	1.0000	0.9375	0.9677
25	1.0000	1.0000	1.0000	26	1.0000	1.0000	1.0000
27	1.0000	1.0000	1.0000	28	1.0000	1.0000	1.0000
29	0.9643	1.0000	0.9818	30	0.9444	1.0000	0.9714
31	1.0000	0.8947	0.9444	32	0.9565	1.0000	0.9778
33	0.9167	0.8148	0.8627	34	1.0000	1.0000	1.0000
35	1.0000	1.0000	1.0000	36	1.0000	1.0000	1.0000
37	1.0000	1.0000	1.0000	38	1.0000	0.9286	0.9630
39	0.9600	1.0000	0.9796	40	1.0000	1.0000	1.0000
41	1.0000	0.9630	0.9811	42	1.0000	1.0000	1.0000
43	1.0000	1.0000	1.0000	44	1.0000	0.9167	0.9565
45	0.9091	1.0000	0.9524	46	1.0000	1.0000	1.0000
47	1.0000	1.0000	1.0000	48	1.0000	1.0000	1.0000
49	0.9615	1.0000	0.9804	50	1.0000	1.0000	1.0000
51	1.0000	0.9474	0.9730	52	0.8261	0.9500	0.8837

(Continued)

Table 4: Continued

Manuscript ID	Precision	Recall	F-score	Manuscript ID	Precision	Recall	F-score
53	1.0000	1.0000	1.0000	54	1.0000	1.0000	1.0000
55	0.9167	1.0000	0.9565	56	1.0000	1.0000	1.0000
57	1.0000	1.0000	1.0000	58	1.0000	1.0000	1.0000
59	1.0000	1.0000	1.0000	60	1.0000	0.9565	0.9778
61	1.0000	0.9375	0.9677	62	1.0000	0.9375	0.9677
63	1.0000	1.0000	1.0000	64	0.9500	1.0000	0.9744

From Tab. 4 we notice that (38) manuscripts were including 1.0000 under their evaluation parameters, which means that they were 100% successfully classified. Moreover, we notice that the lowest recorded F-score that takes into account the calculation of both the precision and the recall was by the second manuscript as 85.11%. Hence, we admit the effectiveness of the VGG19 deep learning model in classifying the images successfully.

4.3 Similarity Measurement and Image Retrieval

To measure the similarity of the classified images, we experimented two developments for the Siamese model. One is the development of the Siamese model from scratch, and the other is its development from the VGG19 pre-trained deep learning model. Considering that both developed models were using the Euclidean (L2) distance metric to measure the similarity scores. The accuracy, precision, recall, F-score, and the mAP evaluation parameters summarized in Tab. 5, were computed to assess the performance of the two developed models.

Table 5: Evaluating the developed Siamese deep learning models

	Built and trained from scratch	Built from a pre-trained deep learning model
Accuracy	0.6171875	0.9771875
Precision	0.6190476	0.9790476
Recall	0.6093750	0.9093750
F-score	0.6141732	0.9141732
mAP	0.5725446	0.9725446

From Tab. 5 we notice that the resulted evaluation parameters were higher when we used the pre-trained deep learning model to build the Siamese deep neural network. Hence, we will use the structure and the weights of the VGG19 pre-trained deep learning model that was initially trained on "ImageNet"[1] dataset images to develop our Siamese network.

After reaching that the Siamese deep learning model performs better when it developed using the architecture of a pre-trained deep learning model, we aim to improve its performance further. Therefore, we experimented using the Siamese model in conjunction with different distance metrics than the Euclidean metric. We tried both the Manhattan (L1) and the Cosine distance metrics and summarized the results in Tab. 6.

Table 6: Siamese deep learning model combined with three distance metrics

	Manhattan (L1)	Euclidean (L2)	Cosine
Accuracy	0.9508929	0.9771875	0.9937500
Precision	0.9697674	0.9790476	0.9750000
Recall	0.9308036	0.9093750	0.9187500
F-score	0.9498861	0.9141732	0.9500000
mAP	0.9372612	0.9725446	0.9937313

From Tab. 6 we notice that the Manhattan distance metric generated 95% accuracy and the Euclidean distance metric generated around 97% accuracy. On the other hand, the highest evaluation parameters generated using the Cosine distance metric. Thereby, we will use the Cosine distance metric to measure the similarity scores of the retrieved images. Fig. 2 highlights the generated confusion matrix by the Cosine distance metric.

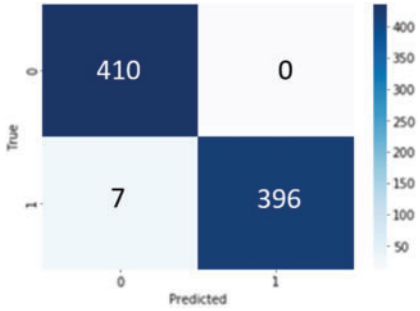

Figure 2: Confusion matrix of cosine distance metric

Since the Siamese deep learning model is a binary classification problem that either classifies the dataset images as similar or as non-similar. Then, the generated confusion matrix in Fig. 2 is including the true positive, false positive in the first row, and the false negative, true negative in the second row. Considering that, our goal is to have high true positives, as well as, high true negatives, to claim that the model is performing good in classifying the images.

From the confusion matrix generated using the Cosine distance metric, we notice that the images classified as true positive were (410), which is an excellent performance. At the same time, the model is performing good in predicting the false negative, and this performance is due to the 99% accuracy. A clear diagonal including the large numbers is created inside the confusion matrix, which confirms the effectiveness of the developed Siamese deep learning model when combined with the Cosine distance metric in classifying the images. Thus, we decided to use the Cosine distance metric in the proposed image retrieval system, and we generated its Precision-Recall curve as illustrated in Fig. 3.

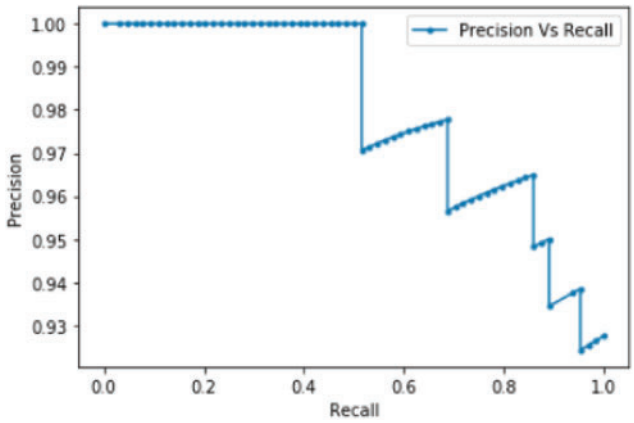

Figure 3: Precision-recall curve of the cosine distance metric

The precision-recall curve helps in evaluating the binary classification problems. Thus, it is a good assessment tool for our problem. The curve summarizes the various generated probabilities among different discrimination threshold values.

From Fig. 3 we notice a steady improvement in the precision values as much as the recall values increase, which reflects the good learning ability of the model using the Cosine distance metric in conjunction with the Siamese deep learning model reaching both precision and recall values that are above 99% successful classification of images.

Considering that the precision-recall curve doesn't reflect the true negatives existed in our confusion matrices. Then, it didn't show the non-similar images that were classified correctly, and thereby, we also generated the Receiver Operating Characteristic (ROC) curve of the Cosine distance metric, as illustrated in Fig. 4.

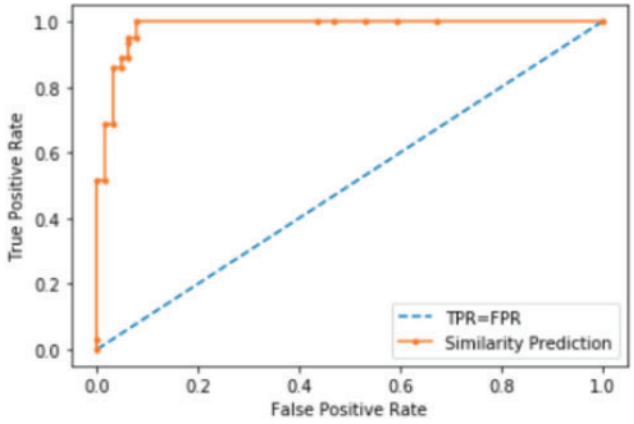

Figure 4: ROC curve of the cosine distance metric

The orange dots within the ROC curve are representing the generated probability values by the Siamese deep learning model using the Cosine distance metric. While the blue dashed line appears in the middle of the curve is to easily visualize the values when the true positive rate equals the false positive rate. From Fig. 4 we notice that the true positive rates are increasing, reaching a high rate that is close to (1). This result reflects the 99.37% of the predictions classified as true-positive rates. In other

words, most of the dataset images correctly classified as either similar or not similar. Whereas, there are few images classified as false-positive rates, which is good because there are only few false alarms generated by the dataset.

However, the model spent many hours training the Siamese model. And even after the training, it consumed long time to generate the output results, and that is due to the Siamese nature in taking two input images, which increased the dataset size exponentially.

Therefore, we installed and activated both CUDA toolkit and cuDNN deep neural network library on our machine to run the code on the GPU instead of running it on the CPU. Considering that the installed NVIDIA GPU on our machine is "Gefore RTX 2080". Thus, its compatible with (10.0.130) CUDA driver and cuDNN version (7). After compiling and executing the same code using the GPU, we reached good enhancements on the results, as illustrated in Tab. 7.

Table 7: Comparison between the Siamese execution on the CPU and GPU

Comparison criteria	CPU	GPU
Total training time	11:53 h	6 h
Image retrieval time	57 s	21.406 s

We notice from Tab. 7 that the overall training time for the ten learning cycles was around 12 h using the CPU. While, it took half of the time using the GPU. Thus, we can save lots of time utilizing a rigid GPU. Regarding the retrieval time of the top-10 similar images to one query input image, there is a dramatic decrease in computing it using the GPU.

Considering that, there is a tremendous number of methods used for image retrieval ranging from the classical static formulas to machine learning, and evolving to reach the deep learning methods. Thereby, we compare in Tab. 8 our proposed method with the very similar approaches to us.

Table 8: Relative comparison with the state-of-the-art methods

Reference, Year	Approach	Distance metric	Dataset	Results
Mehmood et al. [2], 2020	VGG-16 deep Siamese neural network	Softmax	OASIS	**99.05%** accuracy
Tian et al. [3], 2020	Siamese tracking model	Enhanced loss function	OTB2015	60% AUC
An-Bing et al. [4], 2021	Siamese near-infrared spectroscopy model	Euclidean	1314 manually collected drugs images	97% accuracy
Baraldi et al. [5], 2015	Siamese deep learning	ReLU activation function	ImageNet and Places	64% accuracy
Li et al. [6], 2019	ResNet-driven Siamese tracker model	Cross Correlation	TrackingNet	73.3% AUC

(Continued)

Table 8: Continued

Reference, Year	Approach	Distance metric	Dataset	Results
Chen et al. [7], 2019	DSMS-FCN	Absolute difference value	GF dataset	97.89% accuracy
Chaudhuri et al. [8], 2019	Siamese graph convolution network	Euclidean	UC-Merced and PatternNet	69.89% and 81.79% mAP on UC-Merced and PatternNet, respectively
Kande et al. [9], 2021	Generative deep Siamese model	Siamese and restoration loss functions	SDOCT images	68% TP
Radenovic et al. [10], 2018	VGG with generalized mean pooling layer and Siamese model	Euclidean	Oxford5k and Paris6k	91.9% mAP
Zhou et al. [11], 2019	SBLBP based on Siamese multi-layer perceptron network	Intersection Over Union (IOU)	NTU and sketch dataset from Eitz et al.	60% precision on NTU dataset and 39% Precision on the sketch dataset
Koch et al. [12], 2015	Siamese neural Network	Manhattan	Omniglot	93.42% accuracy
Ong et al. [13], 2017	VGG16-Siamese with fisher vector model	Euclidean	Oxford and Paris images	81.5% and 82.5% mAP on Oxford and Paris datasets, respectively
Qiu et al. [14], 2018	ResNet-Siamese deep learning model	Euclidean	TUM	87.7% Precision
Wiggers et al. [15], 2018	AlexNet-Siamese deep learning model	Euclidean	Tobacco800 public dataset	94.4% mAP
Proposed method, 2021	VGG19-Siamese deep learning model	Sigmoid + Cosine	8638 collected Arabic manuscripts' images	**99.37%** mAP

The comparison in Tab. 8 is according to the studies that utilized the Siamese deep learning models as an image retrieval approach. We listed the distance metric and the used dataset as primary comparison criteria. The comparison is relative because almost all of the studies listed in Tab. 8 used existing trending clear images. On the other hand, we collected our dataset manually from text-based faded ink historical images. The highest recorded results were highlighted in bold font style.

From Tab. 8 we notice that the proposed method is generating high image retrieval results without the need for any image segmentation process. That is due to the employed learning strategy, as well as, due to the use of the pre-trained deep learning model to perform the classification and then minimizing the distances among the classified images utilizing the Siamese model. In contrast, we notice that some of the other state-of-the-art methods were requiring segmenting the images, building the models from scratch, or were using irrelevant distance metrics, which lower the image retrieval performance. The research by Mehmood et al. [2] generated close results to ours. However, their result is slightly lower in performance, which might be due to utilizing the VGG16 version of the VGG CNN that is including fewer layers. As well as, they didn't combine the Siamese activation function with any other static distance metric. Therefore, we admit that we reached a novel approach that proves its success, among other existing methods used for image retrieval.

5 Conclusion

This study aims to optimize the image classification and retrieval task, especially for the historical low-quality text-based images. Thus, we experimented the development of the Siamese deep learning model and training it from scratch and compared it with the development of the Siamese model leveraging the weights and the structure of a pre-trained deep learning model. Moreover, we experimented three different distance metrics looking for the most accurate method working in conjunction with the final "Sigmoid" activation function included within the Siamese model to measure the similarity scores. The three-distance metrics are Manhattan, Euclidean, and Cosine. To evaluate the proposed method, we computed the accuracy, precision, recall, F-score, and the mAP.

After extensive experimentations, we found out that building the Siamese model according to the architecture of the VGG19 pre-trained deep learning model is performing better than building it from scratch. In addition, we concluded that the Cosine distance metric was the most accurate method in computing the similarity scores combined with the "Sigmoid" activation function.

Even though training the Siamese deep learning model might be time-consuming due to its nature in comparing two images from the same dataset, it is noticed that the model retrieves similar images quickly after it is trained. Hence, we admit that the proposed model presents a successful solution for solving the classifying and the retrieval of the most similar images to a user query image. The solution worked well with the historical ancient text-based images. Thus, we expect that employing the proposed model with modern clear dataset images, will also generate successful results.

Funding Statement: The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4400271DSR01).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu *et al.,* "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[2] A. Mehmood, M. Maqsood, M. Bashir and Y. Shuyuan, "A deep Siamese convolution neural network for multi-class classification of Alzheimer disease," *Brain Sciences*, vol. 10, no. 84, pp. 1–15, 2020.

[3] S. Tian, X. Liu, M. Liu, S. Li and B. Yin, "Siamese tracking network with informative enhanced loss," *IEEE Transactions on Multimedia*, vol. 23, pp. 120–130, 2020.

[4] Z. An-Bing, Y. Hui-Hua, P. Xi-Peng, Y. Li-Hui and Y. C. Feng, "On-site identification of counterfeit drugs based on near-infrared spectroscopy Siamese-network modeling," *IEEE Access*, vol. 9, pp. 3195–3206, 2021.
[5] L. Baraldi, C. Grana and R. Cucchiara, "A deep Siamese network for scene detection in broadcast videos," *Association for Computing Machinery (ACM)*, vol. 23, pp. 1199–1202, 2015.
[6] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing *et al.,* "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, California, pp. 4282–4291, 2019.
[7] H. Chen, C. Wu, B. Du and L. Zhang, "Deep Siamese multi-scale convolutional network for change detection in multi-temporal VHR images," *IEEE Transactions on Image Processing*, arXiv:1906.11479, vol. 1, pp. 1–13, 2019.
[8] U. Chaudhuri, B. Banerjee and A. Bhattacharya, "Siamese graph convolutional network for content based remote sensing image retrieval," *Computer Vision and Image Understanding*, vol. 184, pp. 22–30, 2019.
[9] N. A. Kande, R. Dakhane, A. Dukkipati and P. K. Yalavarthy, "SiameseGAN: A generative model for denoising of spectral domain optical coherence tomography images," *IEEE Transactions on Medical Imaging*, vol. 40, no. 1, pp. 180–192, 2021.
[10] F. Radenovic, G. Tolias and O. Chum, "Fine-tuning CNN image retrieval with no human annotation," *Transactions on Pattern Analysis and Machine Intelligence (Tpami)*, arXiv:1711.02512, vol. 2, pp. 1–14, 2018.
[11] W. Zhou and J. Jia, "A learning framework for shape retrieval based on multilayer perceptrons," *Pattern Recognition Letters*, vol. 117, pp. 119–130, 2019.
[12] G. Koch, R. Zemel and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. of the 32nd Int. Conf. on Machine Learning, Lille*, France, vol. 37, pp. 1–8, 2015.
[13] E. Ong, S. Husain and M. Bober, "Siamese network of deep fisher-vector descriptors for image retrieval," *Computer Vision and Pattern Recognition*, arXiv:1702.00338, vol. 1, pp. 1–12, 2017.
[14] K. Qiu, Y. Ai, B. Tian, B. Wang and D. Cao, "Siamese-ResNet: Implementing loop closure detection based on siamese network," *IEEE Intelligent Vehicles Symposium*, Changshu, China, pp. 716–721, 2018.
[15] K. L. Wiggers, A. S. Britto, L. Heutte, A. L. Koerich and L. S. Oliveira, "Image retrieval and pattern spotting using siamese neural network," in *Int. Joint Conf. on Neural Networks (IJCNN2019)*, Budapest, Hungary, pp. 1–8, 2019.
[16] M. Khayyat and L. Elrefaei, "Manuscripts image retrieval using deep learning incorporating a variety of fusion levels," *IEEE Access*, vol. 8, pp. 136460–136486, 2020.
[17] H. Lamba, "One shot learning with Siamese networks using Keras," 21-Jan-2019 [Online] *Medium*. Available at: https://towardsdatascience.com/one-shot-learning-with-siamese-networks-using-keras-17f34e75bb3d, 2019, (accessed 04 Oct, 2020).
[18] L. Liu and H. Qi, "Learning effective binary descriptors via cross entropy," in *IEEE Winter Conf. on Applications of Computer Vision (WACV)*, Santa Rosa, CA, USA, pp. 1251–1258, 2017.
[19] P. Singh, "Siamese network keras for image and text similarity," 24-Aug-2019 [Online] *Medium*. Available at: https://medium.com/@prabhnoor0212/siamese-network-keras-31a3a8f37d04, 2019, (accessed 02 Feb, 2020).
[20] S. Guo, Y. Liu, R. Chen, X. Sun and X. Wang, "Improved SMOTE algorithm to deal with imbalanced activity classes in smart homes," *Neural Processing Letters*, vol. 50, pp. 1503–1526, 2019.
[21] M. Khayyat and L. Elrefaei, "Towards author recognition of ancient Arabic manuscripts using deep learning: A transfer learning approach," *International Journal of Computing and Digital Systems*, vol. 9, no. 4, pp. 1–18, 2020.