IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 January 2023, accepted 12 March 2023, date of publication 20 March 2023, date of current version 24 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3259737

RESEARCH ARTICLE

Neural Textual Features Composition for CBIR

MOHAMED A. ABOALI, ISLAM ELMADDAH, AND HOSSAM E. ABDELMUNIM
Computers and Systems Engineering Department, Faculty of Engineering, Ain Shams University, Cairo 11517, Egypt
Corresponding author: Mohamed A. Aboali (g18092914@eng.asu.edu.eg)

ABSTRACT Content Based Image Retrieval, CBIR, is a highly active leading research field with numerous applications that are currently expanding beyond traditional CBIR methodologies. This paper presents a novel CBIR methodology that addresses these growing demands. Query inputs of the proposed methodology are an image and a text. For instance, having an image, a user would like to obtain a similar one while also incorporating modifications described in text format that we refer to as a text-modifier. The proposed methodology uses a set of neural networks that operate in feature space and perform feature composition in a uniform-known domain which is the textual feature domain. In this methodology, ResNet is used to extract image features and LSTM to extract text features to form query inputs. The proposed methodology uses a set of three single-hidden-layer non-linear feedforward networks in a cascading structure labeled NetA, NetC, and NetB. NetA maps image features into corresponding textual features. NetC composes the textual features produced by NetA with text-modifier features to form target image textual features. Finally, NetB maps target textual features to target image features that are used to recall the target image from the image-base based on cosine similarity. The proposed architecture was tested using ResNet 18, 50 and 152 for extracting image features. The testing results are promising and can compete with the most recent approaches to our knowledge.

INDEX TERMS Computer vision, image retrieval, deep learning, feature extraction.

I. INTRODUCTION

CBIR [49] is a subdivision of Computer vision focused on the image retrieval problem as its core problem, The image retrieval problem, literally unchanged, is retrieving an image or images needed by the user from a large database using a reference query input. Searching large image database looking for an image is a painfully time-consuming process for both humans and machines. Therefore, CBIR techniques are not only required to deliver a different and efficient methodology for image retrieval but also to meet or sometimes exceed users' expectations. CBIR performance depends on it's different types of processing including, but not limited to, query formation, feature extraction, feature composition, and similarity matrices. Each stage is a study itself as it has a great impact on the efficiency of the recalled images, so we will introduce a new methodology to improve the efficiency of image retrieval process in its different stages.

The progress in Computer vision was energized by recent technological advances in storage, processing, and data transfer. Computer vision could be seen as a transformation from digital images to digital descriptors that make sense to computer-automated systems to elicit appropriate action. Therefore, unconventional storing-recalling of images is one of the main foundations for computer vision progress [2]. One core problem in CBIR is expressing the needs as it refers to conceptual features formed in the user's mind or an AI machine's conceptual state from different sources and in difficult unknown form. Therefore, researchers in the field provided a variety of means that include drawings, text captions, similar images, icons, sketches, or any combination of them to express that need.

Comparing images is not pixels by pixel-colors based process as these are affected dramatically by insignificant many factors to image-objects or seen-interaction. These factors such as scale, rotation, translation, illumination, pose, orientation, displacements, and many other factors. Consequently, image similarities are applied to features extracted from images and/or image regions. Features are high-level abstractions of image contents. Effective image query formulation is more complicated, in general, than data predicates as images or images abstraction is one of the

The associate editor coordinating the review of this manuscript and approving it for publication was Mingbo Zhao.

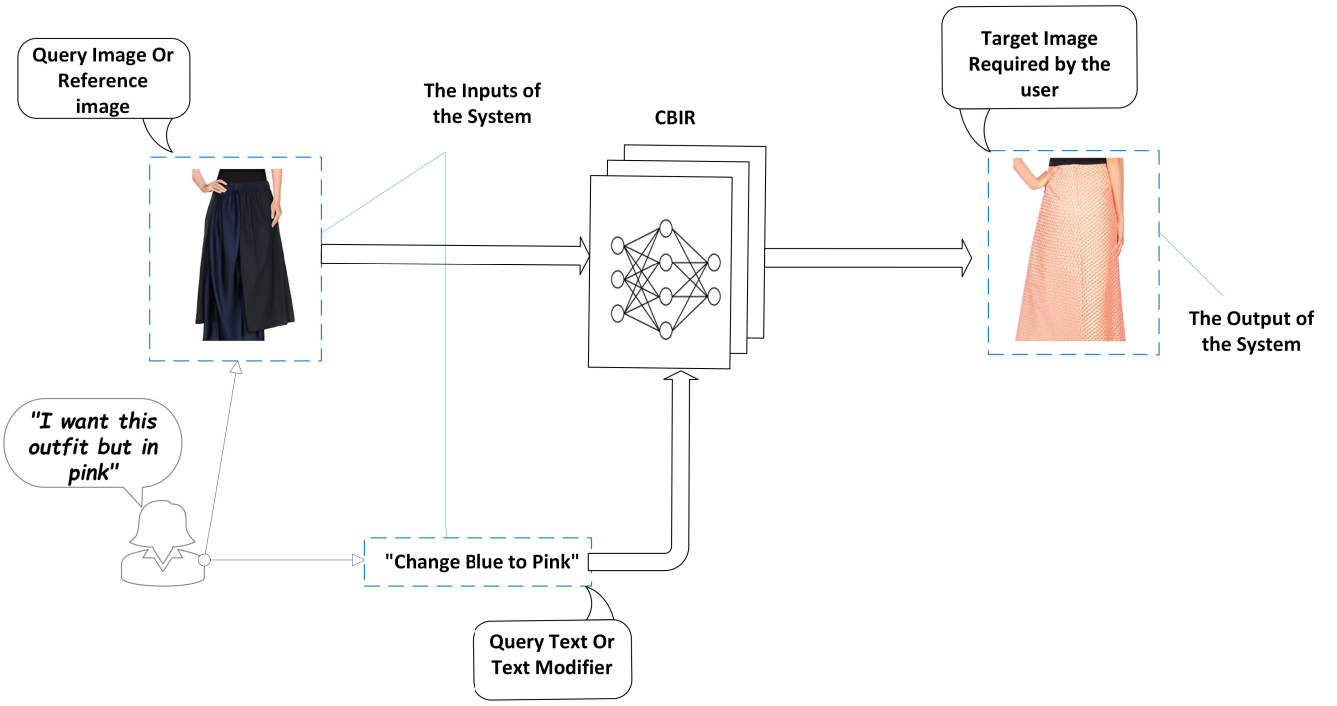


FIGURE 1. Representing the problem definition we face in this study, The user provides us with a reference image and text addressing his/her desired modification on the reference image, the output of the system can be one image or more, and text can represent changes in color or outfit design or material category.

potential inputs. Moreover, texts are likely used with that. The recalling, in this context, contains a high level of uncertainty and possible multiple recalls, or recall-cycles.

Content Based Image Retrieval, CBIR, uses features related to image objects or scenes (an image-objects-interaction). Consequently, CBIR implies changes in image storage and manipulations to enable its recalling facility. CBIR applications include fashion, graphic design, games, simulations, publishing, advertising, historical surveys, architectural engineering, crime prevention, medical diagnosis, geographical information, and remote sensing systems [5].

A typical image retrieval application example is in the clinical decision-making process, it is critical to find other images of the same modality, the same anatomic region of the same disease. Clinical decision support techniques such as case-based reasoning or evidence-based medicine can even produce a stronger need to retrieve images that can be valuable for supporting certain diagnoses. It could even be imagined having Image-Based Reasoning (IBR) as a new discipline for diagnostic aid. Besides diagnostics, teaching, and research are especially expected to improve using visual access methods as visually interesting images could be chosen and found in the existing large repositories. Another example is an online fashion store, the user doesn't want to go through the website or use a simple text search to achieve their target. With CBIR, users can expect a dialogue similar to the one they might have with a salesman in a store that uses text and images similar to the image retrieval recalling process. Fig. 1. Represent an overview of the problem we face in this study.

In this paper, CBIR methodology is proposed to retrieve target image using a query image modified by user text. The proposed approach is based mainly on neural networks to perform feature extraction from images and the modifier text descriptors. The networks used in feature extraction are a) deep convolutional neural network, ResNet, for images and b) recurrent neural networks, LSTM [7], for text. These networks proved to have a significant ability to extract features with high discrimination abilities [6], [7], [8]. The last hidden layer of these networks, the fully connected layers, is replaced with a joint dual-modal net. The proposed methodology maps the dual-modal features to unimodal composed features. Then, the composed feature is mapped to target-image features that will be used for image recall, the following section will go through CBIR stages and how we relate to each one.

We have made two key contributions in our proposed CBIR methodology. The first contribution is in the stage of Feature Extraction, where we introduce the use of different feature extraction neural network architectures, specifically ResNet 18, 50, and 152, to evaluate the new composition process against the comprehensive set of features extracted from those different architectures. The second contribution is in the stage of Feature Composition, where we introduce a novel composition methodology that performs composition in a uniform-known domain, specifically the textual feature domain. This allows for a more consistent and efficient method of combining image and text features for queries.

The rest of the paper is organized as follows. Section II is an overview of CBIR. Section III contains related work followed

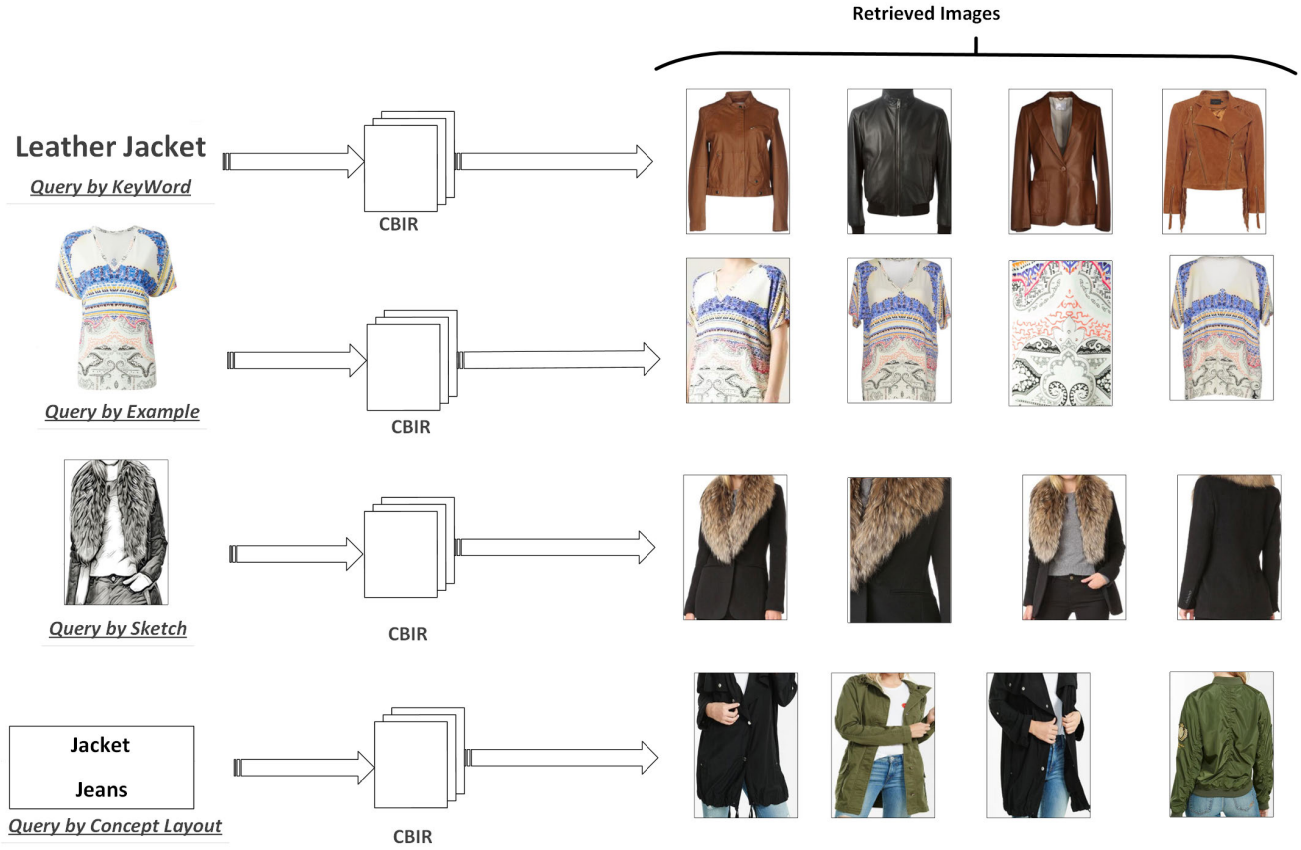


FIGURE 2. Different examples, but not limited to, of query formation used in content based image retrieval, Query formation may be driven by application nature needs but choosing the perfect one will effect the CBIR performance.

by section IV where we introduced the proposed model. Tests and results are presented in section V and finally, section VI presents our conclusion.

II. CONTENT BASED IMAGE RETRIEVAL

The term "Content-Based" implicitly requires content comprehension that mandates analysis and prior knowledge contrary to using metadata such as keywords and descriptions associated with images. "Content" of images implicitly refers to colors, shapes, textures or even objects interactions inferred from images. Field researchers refer to information extracted to represent these contents by features. Query formation, feature extractions, a combination of features and distance metrics are major CBIR factors [29], [30]. These topics are discussed briefly in the following subsections, Fig.3 shows an overview of CBIR Process.

A. QUERY FORMATION

Query formation is driven by application needs. Applications specify query inputs as well as the form of data or information available for the recall process. The CBIR queries are significantly difficult as it relates to abstract hypothetical concepts, images or thoughts formed in users' mind. Such query expression requires widening how the formulation takes place. Therefore, researchers use varieties of types for query formation that include images, keywords, sketches, color maps, canvases, context maps, as well as icons [31]. Query by Keywords suffers from ambiguity and there could be a significant gap between the image and its label [32]. Moreover, the annotation process is painful, likely incomplete, and lakes precision. Another type of query formation is query by example (QBE): in which attributes or features, hopefully, describe the contents of the user's desired image [33], [34]. In query by canvas, users use geometrical shapes, colors, and textures, the canvas expresses the needs using primitive features [33], [34], [35]. Query by spatial icons uses higher-level visual semantics to represent spatial arrangements or interactions of images or image segments. Query formation using text and/or example image is used in [25], Fig.2 Shows some examples of query formation.

In this paper, our query formation consists of two query inputs, a reference image and a text modifier, The text describes the user requested modification on the reference image to match with his target image, the use of different query inputs was always a case study, combinations of the formerly mentioned formations are used in query formation by [32], [37], and [38]. The existence of more than one query input will argue the need for adding a feature composition stage later on.

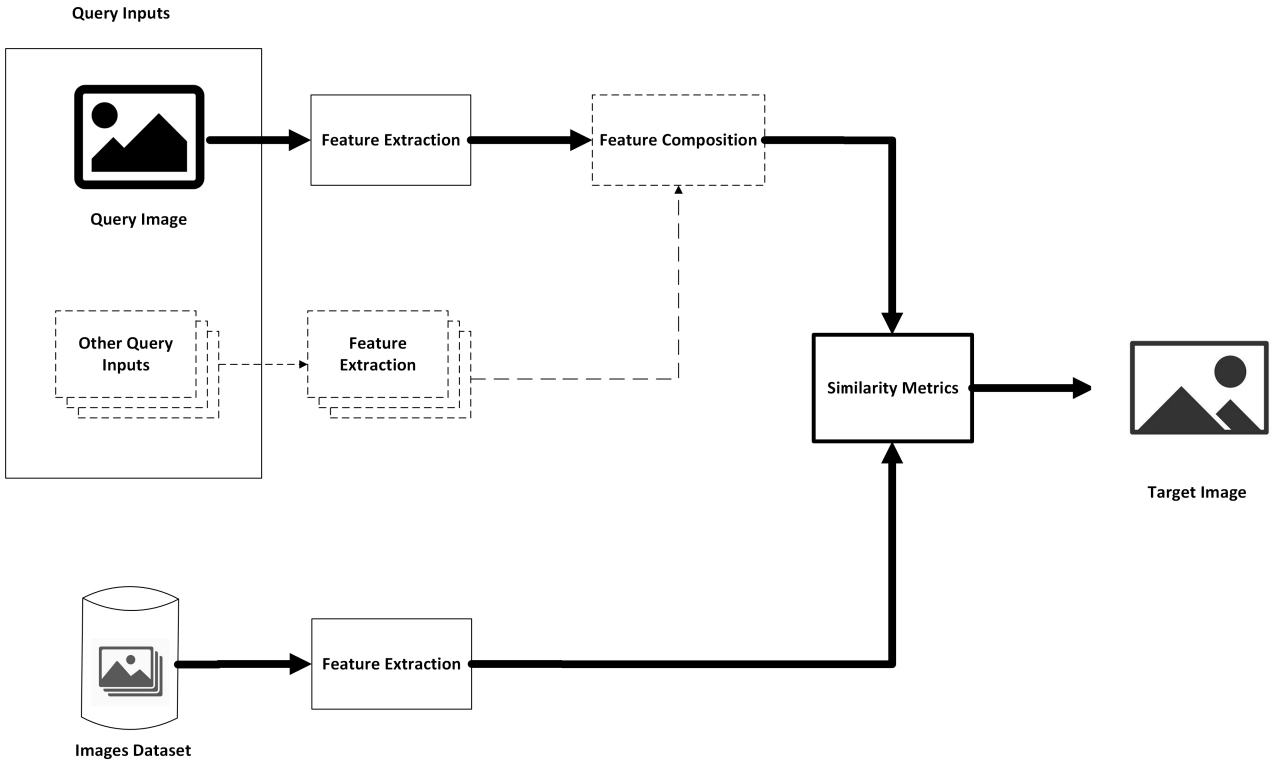


FIGURE 3. Representing the full cycle of the image retrieval process Including query formation which may contain more than one query object, feature extraction for both query and target objects, feature composition of all inputs features and similarity metrics matching required target image. The dashed lines represent functions that may not exist in some CBIR architectures.

B. FEATURE EXTRACTION

In machine learning feature extraction, machines are trained to select and extract the features using a training dataset. The main machine used for this task is the artificial neural network, ANN [20]. In a neural network, the designer engineers' network and selects the proper training dataset to suit the task. Then, Network is trained and validated using the prepared training dataset. The process seems to be easier than the engineered feature extraction approach. however, there are many questions the designers must answer such as: which network architecture to use? How many layers? How many neurons are in each layer? What is the proper training dataset? What is the target of the learning process required to reach? Which nonlinear function to use? and what is the validation dataset?

Feature extraction could be engineered, or machine-learned. In the case of engineered features, designers study the possible features. Moreover, an experimental study is likely included over a sample set to specify a set of features suitable for the application. This process is called feature selection. Then, an algorithm designed for the extraction process of the selected features is applied. This engineered approach assures that the selected features carry the desired properties which ensure insignificant distances for same-class image concepts and vice versa. Moreover, it assures small variance for scale, rotation, noise, pose, and translation on extracted features. The algorithms-selection takes into consideration the computational complexity to assure minimal computational cost [39]. In machine learning feature extraction, machines are trained to select and extract the features using a training dataset. The main machine used for this task is the Artificial Neural Network, ANN [18].

Feature vectors contain measurements of different attributes of images or image objects. Therefore, feature vectors, normally, requires preprocessing before use due to the heterogonous nature of their elements. Standardization and normalization could be part of the extraction process for both engineered or learned cases. Also, it's intuitive to include enhancement steps such as smoothing, sharpening, and contrast adaption ahead. Images with dominating background pixel counts will not be suitable for direct applications to features' operators therefore background removal and other preprocessing techniques are recommended.

Feature learning or automatic representation is embedding machines with a basic set of techniques. These techniques allow systems to automatically discover features from raw data. The engineering process herein is for feature learning rather than extracting itself. Feature learning is motivated by the fact that machine learning, in general, mandates feature extraction steps ahead. Therefore, the integration of the two steps is mathematically and computationally convenient. Also, for real applications features extraction from data, such as images, video and sensors, are not algorithmically deterministic. Also, features that suit one application may not

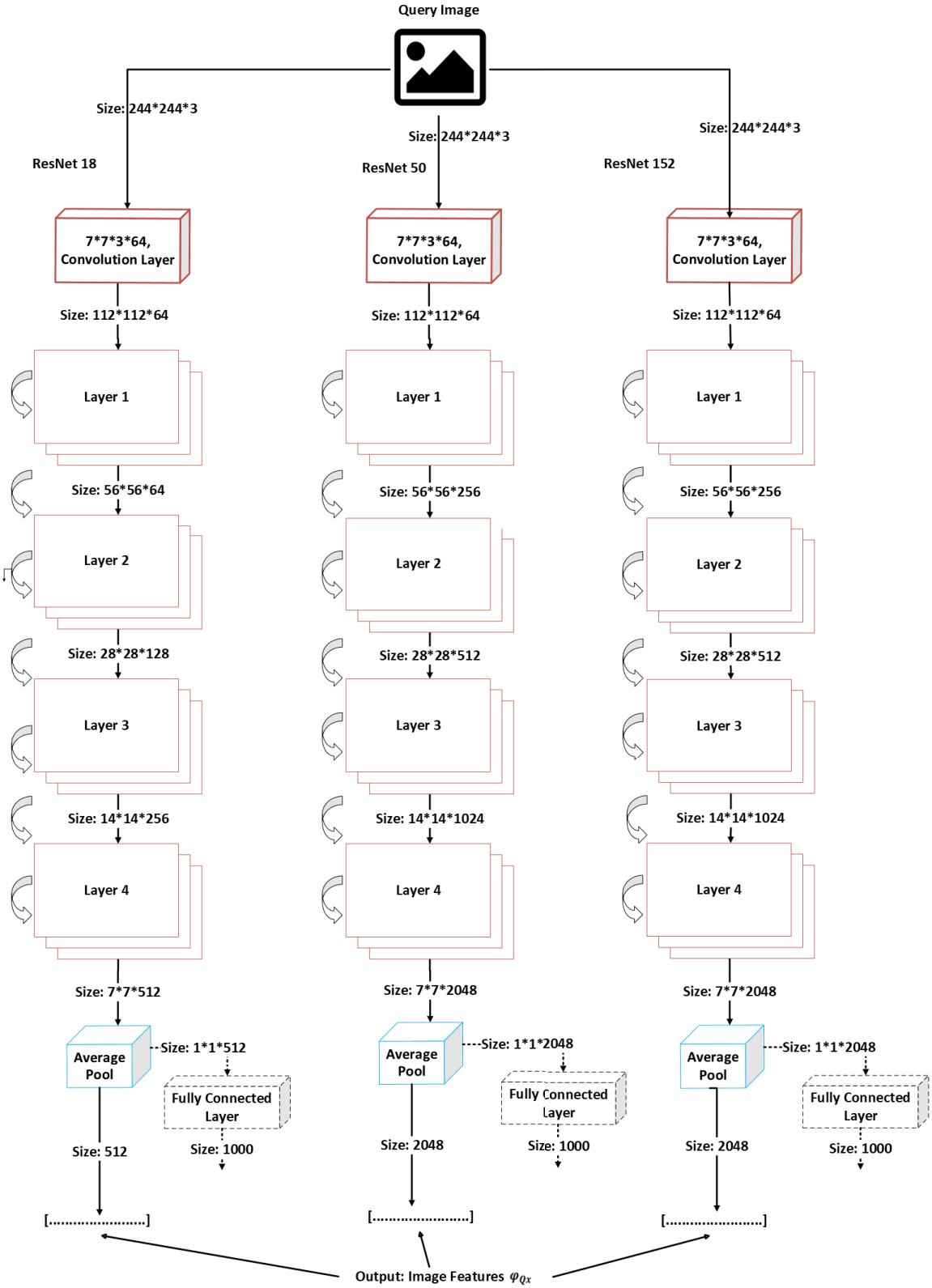


FIGURE 4. Representing 3 different feature extraction used in this study, 3 pre-trained models with modification to extract the last convolutional layer in each model, For ResnNet 18 size of the features will be 512, ResNet 50 and ResNet 152 is 2048.

suit the other. An alternative road is to discover such features or representations through the trial-assessment learning process, without relying on explicit algorithms. Feature learning can be either supervised or unsupervised. Supervised

learning requires training and validation data to be labeled. From the supervised networks perceptron, radial bases, and Convolution neural networks. In unsupervised learning [8], features are learned without the need for labels. From the unsupervised architectures networks, Kohonen network, self-organizing map and Hopfield network.

Features could be extracted either in a local or global fashion. In global features extraction, the feature operator applies to the whole image. Local feature extraction is regional to the image. For the search of any structured data, specific techniques like convolutional methods using hand-crafted kernels or syntactic and structural methods are employed. These techniques encode problem-specific knowledge into the features. The use of such methods led to significant improvement in object recognition and machine learning [39]. Neural networks used in feature extraction include feed-forward and recurrent networks both shallow and deep [27], [28], [29].

In this study, image and text-modifier are presented by the extracted features. Image features extraction techniques used are ANN-CNN deep-networks, specifically, ResNet. Most famous architectures ResNet 18, 50, 152 as in Fig.4, are included in this study for sake of comparison and studying their effects on the quality of the retrieval process. The models used are image-net pre-trained models. The fully connected layer input vector is considered the feature vector. The text features are extracted by the well-known RNN-LSTM net with a feature vector cardinality of 512.

C. FEATURES VECTORS COMPOSITION

The composition principle, in general, is rooted historically in philosophy, mathematics, neuroscience, as well as many other computer science domains such as natural language understanding, sensor fusion, visual recognition, and many others. In visual recognition, the main principle is to make up a new concept from primitive conceptual elements. One of the main topics using feature combination is object and image classifications as it is needed to improve the classification accuracy compared to the use of a single feature. The composition base, in general, is statistical learning from samples [10]. Researchers work towards generating new concepts from primitive ones focused on building models as well as models training from samples in [9]. The compositionality for visual recognition includes the work of Biederman's Recognition-By-Component's theory [10] and Hoffman's part theory [11] as landmarks.

Feature composition is needed in CBIR in cases when different sources of features exist such as our case of text and image. The composition objective is to generate a new powerful feature that carries both. The multilayer perceptron is used to concatenate text and image features in [40], [41], and [42]. LSTM, as a recurrent model, fed by image features followed by text words is used in [25] and [47]. Text used to form transfer matrices for image features is discussed in [26]. Visual question answering methodologies, that focus on finding answers to natural language questions on a given image were used in [20], [43], [44], and [45].

The use of convolutional Neural Networks as a means of feature representations of multiple semantics is used in [9]. Objects composition for recognition systems uses Deformable Part Models in [12], grammars in [13], and AND-OR graphs in [14]. The composition as the pivotal element is used for visual question answering in [15], handwriting recognition in [16], and zero-shot detection in [17]. A lot of research has been done to improve feature composition retrieval performance by user feedback on relevance. A Multimodal Compact Bilinear Pooling as a feature fusion mechanism to combine image and text was proposed in [19]. In [20], the text feature is incorporated by mapping into parameters of a fully connected layer within the image CNN. Another domain that incorporates text image composition is visual question answering [21]. The residual connection is used to enforce composition in [22]. In [23], the recurrent model to colorize images given text descriptions. In [24], adding text descriptors localize objects within input images. In [25], the composition classifier was trained to combine an object classifier and an attribute classifier. The attribute embedding operator was used in [26].

In this study, feature fusion of features is performed in the textual space. That is between a textual feature of the query image and the text modifier. Therefore, a mapping stage needed a head to translate image features to corresponding textual features.

D. SIMILARITY METRICS

Content-based image retrieval (CBIR) not only needs efficient extraction of features and principles composition but also an effective similarity metric. The similarity metric measures the distance between feature vectors. Features distance metric should take into consideration that feature-vector elements are of completely different distributions and represent apart concepts with different properties. Various distance metrics are used to measure the distance between vectors that include Euclidean distance, city block distance, Canberra distance, maximum value metric, Minkowski distance, Mahalanobis Distance, Histogram Intersection Distance, and Quadratic Form Distance [31]. Distance metrics are characterized by their accuracy, suitability to features, and the time complexity.

III. RELATED WORK

Many approaches are used to resolve the CBIR which is also known as Query Based Image Retrieval (QBIR), such as scale-invariant transform and vector of the locally aggregated descriptor. The great performance results of neural networks manipulating unstructured data inputs such as images and text attracted researchers to propose solutions for many problems based on neural networks. Deep Convolution Neural Networks, DCNN/CNN, manipulate raw images with a prominent performance in many applications such as image classifiers. Neural word embedding based on a continuous bag of words, skip-gram and many others opened the door for text manipulations. Recurrent Neural Networks

architectures such as LSTM and GRU embed text processing abilities of long-term memories attracted researchers to use extensively in many text-processing applications such as translation and review tagging. Moreover, many of the libraries such as Torch, Keras, and TensorFlow contain many pre-trained classifiers neural models on significantly hard datasets such as ImageNet. Those models could be adapted or partially utilized for another application domain using transfer learning. Transfer learning saves time, and reduces cost, and effort in design and validation.

Deep metric learning (DML) is a technique that aims to employ deep neural networks to learn a metric. The learned metric ensures that the distances between samples of the same class are smaller than the distances between the samples of different classes. This learning technique has been employed in building classifiers and as a cross-modal retrieval image retrieval. That is, retrieving images based on text query and getting captions from the database based on the image query [2], [4], [52], [53], [57], and [46].

In Visual Question Answering, DML is used to fuse the text and image inputs [44], [55], and [20]. The relationship is a methodology based on relational reasoning [44]. Image features are extracted from CNN and text features from LSTM to create a set of relationship features. These features are then passed through a MLP and after averaging them the composed representation is obtained.

In FiLM [55], the source image is ''influenced'' by an affine transformation to the output of a hidden layer in the neural network. A parameter hashing-based method [20] where the fully-connected layer in a CNN acts as the dynamic parameter provider. The use of MLP to fuse text and image features has been studied by Vo et al, [25]. They propose a gated feature connection to compose a representation of the query image and text to the target image. They also incorporate residual connection and convolution functions in the learning process to perform the concatenation of image-text features like the target image features. Another simple effective approach is the Show and Tell [27]. They use LSTM, the RNN, to predict the next text word in the sequence of the textual descriptor. The prediction is based on the seen image and the former words. The LSTM final hidden state is considered the composed image-text representation.

Term frequency-inverse document frequency as a description vector based on CNN is proposed for CBIR in [3]. For this purpose, the learned filers of convolution layers of the convolution neuron model were used as a detector of the visual words. The authors conduct experiments on four image retrieval datasets and the outcomes of the experiments show the existence of the truth of the model.

Shi et al. [36] proposed a hashing algorithm that extracts features from images and learns their binary representations. The authors model the pairwise matrix and an objective function with a deep learning framework that learns the binary representations of images. Han et al. [24] approach is to learn spatially aware attributes from product textual descriptions and then use it to retrieve products from the database. The provided text query vocabulary is limited to a predefined set of attributes. Nagarajan et al. [54] proposed ''Attribute as Operator'' which is an embedding approach where text query is embedded as a transformation matrix. The image features are then transformed with this matrix to get the composed representation. The latter approach is closely related to interactive image retrieval [41], [56] and attribute-based product retrieval [25], [28]. These approaches share their limitations to a fixed set of relative attributes [28] and likely require multiple rounds of queries as input [41], [56] or query texts to be only one word, an attribute [24]. In our case, the input query text is not limited to a fixed set of attributes and does not require multiple interactions with the user.

IV. METHOD

To point out the problem which the design treats easily we should go through the standard dataset used in the study. The dataset is Fashion200k [48] which contains 200K+ dataset elements. The dataset splits into a training subset of 172K training and 33k testing elements chosen randomly. The dataset contains images of different classes of fashion items. Each image of the dataset comes with brief text/caption about the image. Dataset queries are created [25], [47] with a dataset image and modifier text containing a target word which replaces a specific word in the query image captions that makes the target image caption. The training and testing datasets are formed in this way which makes it a supervised training dataset. An element in the training and testing datasets same as in Fig. 5.

The proposed methodology shares bases with studies in [1], [25], and [47]. These bases include the query inputs, the use of neural networks, and testing results on the Fashion 200K dataset. The networks used in the formerly mentioned studies include deep, shallow, and recurrent. In these studies, neural networks are used in feature extraction from both images and text modifiers, features composition, and our target, to construct a new feature vector similar to the target image feature extracted by composing both image and text features, so that our target will be to achieve the below equation:

$$cos(f_{combine}(\varphi_{Qx}, \varphi_{QMt}), \varphi_{Tx}) \leq cos(\varphi_{Dx_i}, f_{combine}(\varphi_{Qx}, \varphi_{QMt})) \quad (1)$$

where:

- φ_{Qx} is the query image feature.
- φ_{QMt} is the query modifier text feature.
- φ_{Tx} is the target image feature.
- $\forall i \in$ *the dataset images*
- φ_{Dx} is the dataset image features.

A. COMPARATIVE STUDY COMPOSITIONAL METHODS

In the formerly mentioned studies, image features φ_x were extracted using Resnet 18 pre-trained model. The extracted feature vectors, used, are the output of the last convolution layer of size 512 as in Fig 4. The text features φ_t extracted using an embedding layer followed by an LSTM network

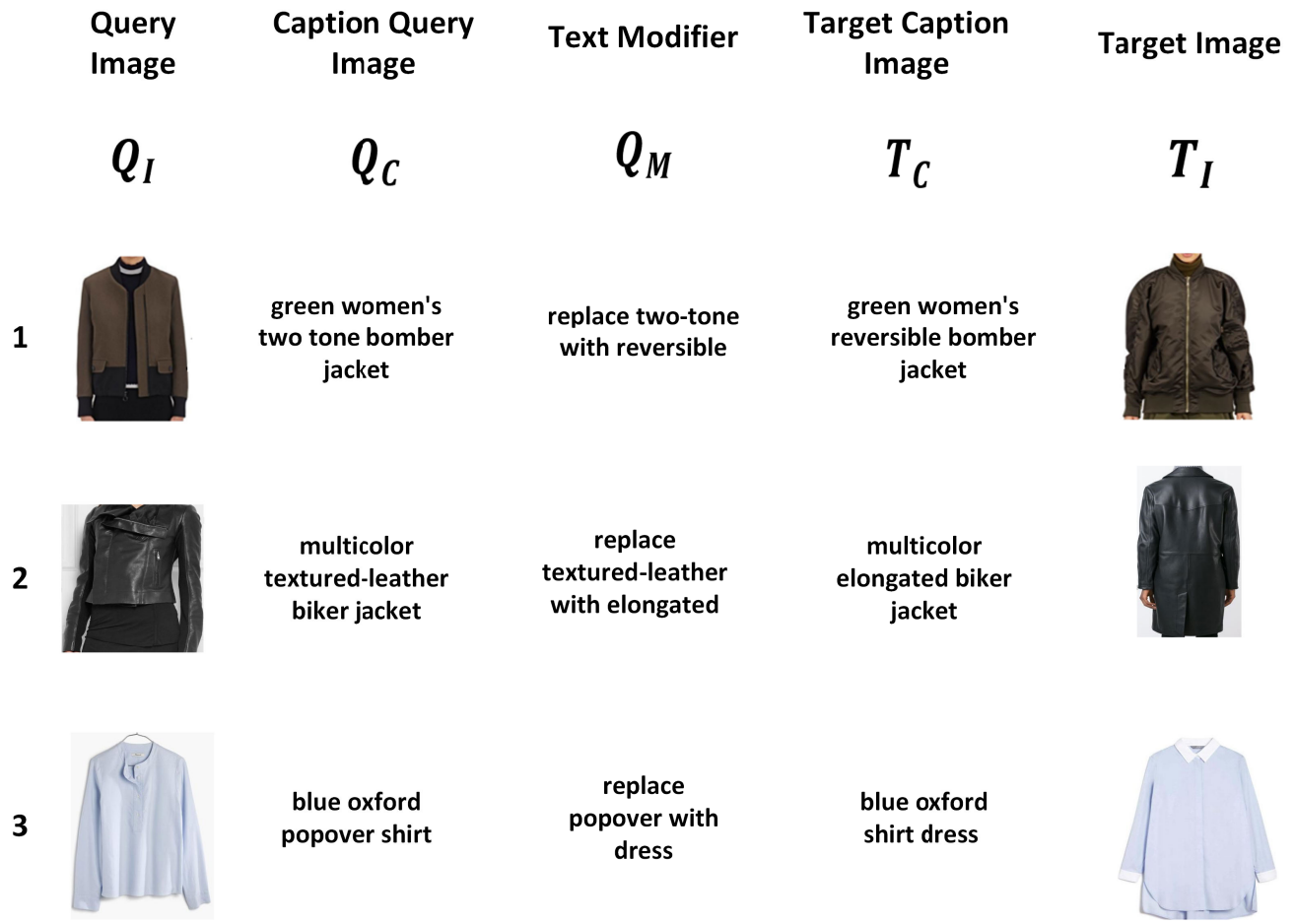


FIGURE 5. Represent 3 random samples from the dataset queries using Fashion200K dataset, each query has five elements, as an example let's take the first one, query image 'Q'_I is an image on the left, query caption 'Q'_C is "green women's two-tone bomber jacket", "replace two-tone with reversible" is modifier string 'Q'_M, target image 'T'_I is image on the right and target image caption 'T'_C is "green women's reversible bomber jacket".

with a hidden vector of size 512. The major problem in the CBIR case at hand is the composition of the feature between the image vector and textual vector to form the recall vector. The treatments for this composition were different in the three studies. Therefore, we will summarize that in the following paragraph.

In [25], the authors created a new function called Text Image Residual Gating, TIRG equation (2), which is used to combine image and text features. The TIRG function is a weighted sum of the residual function equation (3) and gating function equation (4). The weights are supposed to be adapted through the neural learning process. The core of both residual and gating functions is a sequence of convolutions followed by Relu's. Then, the convoluted text features broadcasted on image features. The gating function has two more steps sigmoidal application followed by a dot product with image features. The author intends to "modify" the query image feature instead "feature fusion" to create a new feature from existing ones. Which is represented by the gated identity establishes the input image feature as a reference to the output composition feature, as if they were in the same meaningful image feature space, then the added residual connection represents the modification in this feature space. The function summary is:

$$\varphi_{xt}^{rg} = \omega_g f_{gate}(\varphi_x, \varphi_t) + \omega_r f_{res}(\varphi_x, \varphi_t) \tag{2}$$

where:

- φ_{xt}^{rg} represent TIRG.
- ω_g, ω_r are learning weights.
- φ_x Represent the Last layer of ResNet 18 (ResNet 17) Last convolution layer (H*W*C), W is the width = 1, H is the height=1, and C = 512 is the number of feature channels of image.
- φ_t Represent the last layer of LSTM of text (C = 512).
- $f_{gate}(\varphi_x, \varphi_t), f_{res}(\varphi_x, \varphi_t)$ linear mapping two convolution functions called gating and residual functions

$$f_{gate}(\varphi_x, \varphi_t) = \sigma(W_{g^2} * RELU(W_{g^1} * [\varphi_x, \varphi_t])) \odot \varphi_x \tag{3}$$

$$f_{res}(\varphi_x, \varphi_t) = (W_{r^2} * RELU(W_{r^1} * [\varphi_x, \varphi_t])) \tag{4}$$

where:

- σ is a sigmoid function.

- W_2, W_1 are 3×3 Convolutional Filter.
- $\odot$ is element wise product.

In [1], The TIRG function was augmented by an optimization layer. The optimization layers included in the study are non-linear-MLP and linear regression function. The study concluded to:

- The TIRG composed 512 features vector is abstracted to the level where further optimization leads to Mean Square Error, MSE, enhancement while a big loss in generalization.
- Semantic bridges are needed for textual features as text features relate to words embedded vectors not fully maintaining words' semantic distances.
- Semantic abstraction not recommended at earlier stages. As early loss cannot be recovered at later steps.

One solution to the semantic-related former notes is to establish a relationship between image features and their textual features. Together with concepts compositions are to be done between concepts of the same modal bases. These minimize the side effects of the semantic issues raised in this study.

In [47], features composition using an autoencoder called ComposeAE to compose query muti-modals. Image features were extracted using CNN-ResNet17. Text features were extracted using the BERT model. Training set formed from the query-image features, text-modifier features, and target-image features. The ComposeAE is built upon the assumption that the source image and target image are rotations of each other in some complex space. That is, the target image representation is an element-wise rotation of the representation of the source image in that complex space. The information needed to get the proper rotation of the source is encoded in the query text features.

The text features are modeled as the element-wise rotation of source image features by forming a rotational diagonal matrix as a training target and then training nonlinear-MLP to learn the mapping function. The image features were mapped using another MLP network to an equal-dimensioned domain to present all possible rotations. The text-features-trained network is used to map the output of the second network to form a compositional form of the two features. This compositional output together with the raw features text and image form inputs to the convolution-pooling step. This final stage outcome is used to enforce the optimization mapping function that establishes the query functions using the third-nonlinear-MLP neural network. The network overall function composition and optimization function could be summarized in equation (5).

$$f(q, z) = \rho_1(\tau(\rho_1(\rho_2(z)), q, z)) \tag{5}$$

where:-

- q,z are text and image extracted features.
- ρ_1, ρ_2 are the three Non-linear MLP.
- τ convolution and pooling function.

B. THE PROPOSED METHOD

The proposed architecture, Fig. 6, is composed of three single-hidden nonlinear feedforward fully connected sub-networks NetA, NetB, and NetC. Two mapping networks, NetA, NetB and a compositional network NetC. NetA maps image features to the corresponding textual feature, and NetB perform the later mapping inverse. That is, NetA function maps φ_{Qx} which is the query image extracted by ResNet to φ_{Qt} which is an approximate image caption feature. NetC, the compositional network inputs are two text features: φ_{QMt}, query modifier textual features and φ_{Qt}. Both NetC inputs represent textual features of similar or the same modal bases that likely makes composition effectively and smoothly operational. NetC outcome, φ_{Tt}, which is an approximate target caption feature. The final stage, NetB, input is φ_{Tt}, and the net maps it to possible target image features. Apparently, the three networks could be trained in parallel on perfect data from the dataset.

The proposed methodology aims to combine features in the same space which is the text features. The composition vehicle used is a neural network. The problem solution spaces are known and logically consistent with humans processing for such cases. Homogeneity in feature composition logically enhances the recalling results and makes network generalization likely better. Moreover, in this study different image features extraction methodologies were used using pre-trained ResNet architectures 18, 50, and 152, to review their impact on the quality of retrieving images. Adding those architectures led to three different models for each of NetA and NetB.

To present the logic beyond the proposed architecture. Let us assume that the dataset query of Fig. 5. is presented to a human. One can also easily infer that when we see a fashion model or even any-named object our brain recalls its name or associated caption. This recall step becomes mandatory in case of such query to enable brain textual replacement. Consequently, a new conceptual textual feature was formed for the target. Finally, using the formed new textual feature brain recalls the associated conceptual target image. This conceptual target image is the one we use to look for needed images. These steps cannot take place without well-structured knowledge about image-features mapping to captions-features, caption features composition abilities and caption-features mapping to image-features-concepts. Therefore, we claim it is logically intuitive and semantically problem-fit. In the following sections, we will discuss each model in our architecture.

ResNet 18, 50, and 152, are standard network architectures proven to outperform many other architectures in computer vision competitions [6]. As a well-known deep net and considering that deep learning of such architectures takes a huge number of FLOP transfer learning proved to utilize successfully the pre-trained models feature map for other applications. In our study, A pre-trained ResNet model is used to extract image features (φ_x) from images to operate in a feature space. Moreover, the same models used to produce the features associated to be associated with the image database.

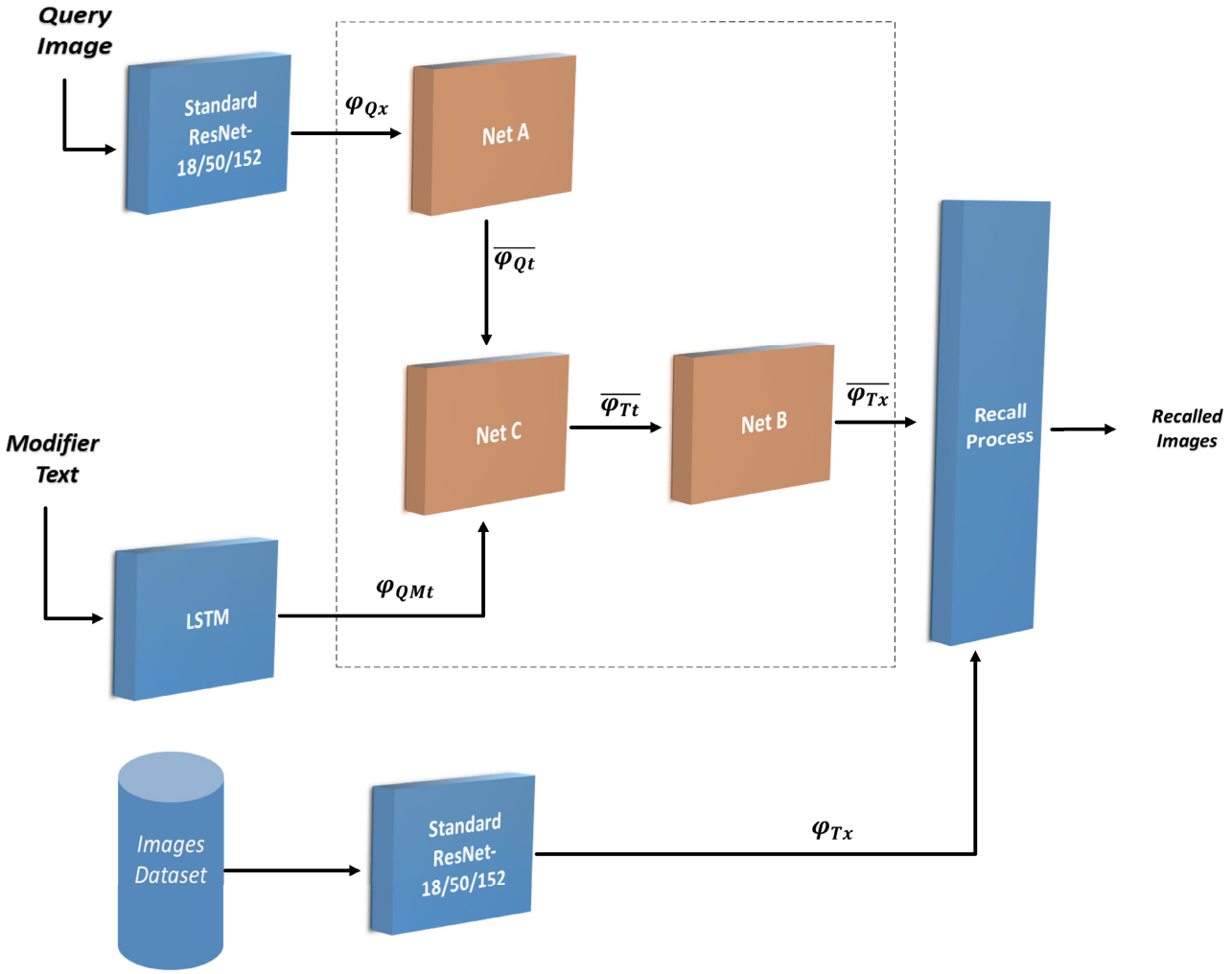


FIGURE 6. The proposed network architecture in the recalling process, Left to the dashed lines is the feature extraction stage, the dashed area represents the 3 proposed nets, NetA maps query image feature to query caption text feature, NetC Combine both query caption text feature and modifier text feature to one feature (the target caption feature), NetB maps combined feature to target image feature to be used in recalling process to find similar image features which are extracted from the dataset of images.

The use of pre-trained models for the same purpose was used in [1], [25], and [47] on a single architecture and differently employed. The extracted features are after the last stage of the convolution layer and before the connected layer. The size of the extracted features of ResNet 18, 50, and 152 are 512, 2048 and 2048 respectively.

LSTM, Long Short-Term Memory model is a recurrent neural network model. The LSTM model is used in text-processing applications with the ability to keep memorizing features for later use. In our model LSTM is used for extracting text features from the text (φ_t) for the query caption features (φ_{Qt}), target caption (φ_{Tt}), and query modifier text (φ_{QMt}). These features are used in the training features compositional network, NetC. Also, during the recalling process, it is used to extract features from the text modifier supplied with the query image. The LSTM network is set to extract a feature vector of size of 512, based on a trained and used in [1], [25], and [47].

NetA is a function mapping image features to associated image caption features. During the training, image features are extracted by the ResNet as input and LSTM extracted features from the image caption as a target. During recall, the query image extracted features by the ResNet will be supplied to NetA to supply NetC with query image caption text features. NetA is a fully connected neural net network with a single hidden layer. The network inputs of this layer (φ_{Qx}) are the size of 512 in the case of ResNet18, 2048 in the case of ResNet 50,152. The output of size 512 represents ($\overline{\varphi_{Qt}}$) text feature of the query caption text. The hidden layer size of the network in our experimentation was 1000 neurons. The 1000 size represents an expansion to compact image features case of 512 close to doubling and intermediate contraction in the other two cases.

NetB represents another mapping layer from text features of the output of NetC to the target image feature of the target image, NetB is, also, a fully connected neural network of the single hidden layer. NetB input vector cardinality is 512 and

the output vector sizes are 512 in the case of ResNet 18, and 2048 in the other two cases. The hidden layer size is set to 2500 during the experimentation to represent an expansion even for the target case of 2048 features.

NetC is trained to combine features of the modifier captions with image caption features to produce target image captions. The operating domains for inputs and output of the network are the same, text features. In the query, operational mode NetC combines both text features of inputs query text caption feature ($\overline{\varphi_{Qt}}$, the output of NetA), a size of 512 and extracted modifier text feature (φ_{QMt}) of size 512. The network output size 512 ($\overline{\varphi_{Tt}}$). NetC is also a fully connected network of a single hidden layer. In our experimentation, the hidden layer size was set at 1800 to represent an expansion for the inputs to enable composition operation.

The recalling process uses the nearest neighbor approach based on the Cosine Similarity metric [50]. The use of cosine similarity is more popular than the Euclidean distance [51] in image feature spaces as it considers the direction of the vectors other than the position of the point in the space. The position of the vector in space contains the feature vector magnitude which affects drastically, in the general case with image intensity offset which is insignificant to image objects interactions.

For more clarity and overall functional specifications of the proposed solution. Let us assume: - η_L, η_R are the LSTM network, and ResNet transfer functions to the inputs flatten vector of input of the FC layer of that network, and τ_A, τ_B, τ_C are the networks NetA, NetB, and NetC transfer functions consequently, the proposed net overall function and algorithm could be summarized as: -

$$\tau_A(\varphi_{Qx}) = \bar{\varphi_{Qt}} = \tanh(\varphi_{Qx} * W_A^1) * W_A^2 \tag{6}$$

$$\tau_B(\bar{\varphi_{Tt}}) = \bar{\varphi_{Tx}} = \tanh(\bar{\varphi_{T}t} * W_B^1) * W_A^2 \tag{7}$$

$$\tau_C(\bar{\varphi_{Qt}}, \varphi_{Qx}) = \bar{\varphi_{Tt}} = \sigma([\bar{\varphi_{Qt}} : \varphi_{QMt}] * W_C^1) * W_C^2 \tag{8}$$

where:-

- σ is the sigmoidal function
- $\varphi_{Qt} = \eta_L(Q_c)$
- $\varphi_{QMt} = \eta_L(Q_M)$
- $\varphi_{Tt} = \eta_L(T_c)$
- $\varphi_{Qx} = \eta_R(Q_I)$
- $\varphi_{Tx} = \eta_R(T_I)$
- W_X^1 net-X output-hidden weight matrix
- W_X^2 net-X output-hidden weight matrix

The three networks were trained using Google-COLAB-pro utilizing the platform's powerful Graphic Processors Units (GPU). The training process of neural networks, in general, is an optimization process that requires a loss function known, also, a loss metric and an optimization model. In our case, the mean square error (MSE) equation (8) for networks NetA and NetC.

$$MSE = \frac{SSE}{n} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{9}$$

Algorithm 1 The Recalling Algorithm

Require: Query Image Q_I
Require: Text Modifier Q_M
Require: Images Dataset T_I
Require: Image Feature Extraction Model η_R
Require: Text Feature Extraction Model η_L
Require: Model NetA, NetC, NetB τ_A, τ_C, τ_B
Require: Similarity Metric Function $f(s)$

$\varphi_{Qx} \leftarrow \eta_R^n(Q_I)$ ▷ n is 18 or 50 or 152
$[\varphi_{Tx}, \ldots] \leftarrow \eta_R^n([T_I, \ldots])$
$\varphi_{QMT} \leftarrow \eta_L(Q_M)$
$\bar{\varphi_{Qt}} \leftarrow \tau_A^n(\varphi_{Qx})$
$\bar{\varphi_{Tt}} \leftarrow \tau_C([\bar{\varphi_{Qt}} : \varphi_{QMT}])$
$\bar{\varphi_{Tx}} \leftarrow \tau_B^n(\bar{\varphi_{Tt}})$
$[T_I, \ldots] \leftarrow f(s)(\bar{\varphi_{Tx}}, [\varphi_{Tx}, \ldots])$

where:-

- n is batch size

For NetB, the cosine similarity metric is more commonly used in image feature comparisons since feature vector magnitude is greatly affected by the image intensity. Which is defined as following:-

Assuming $\odot$ is the DOT operator of two vectors and **v1**, **v2** are n-dimensional vectors. Then, $\mathbf{v1} \odot \mathbf{v2} = ||\mathbf{v1}||||\mathbf{v2}|| \cos(\theta)$, where θ is the angle between the two vectors, $||v||$ is vector magnitude. $\cos(\theta)$: is called the cosine similarity between **v1**, **v2**. The cosine similarity loss metric is defined as: -

$$1 - \cos(\theta) = 1 - \frac{(\mathbf{v1} \odot \mathbf{v2})}{||\mathbf{v1}||||\mathbf{v2}||} \tag{10}$$

The cosine similarity loss function yields 0 when the two vectors are in the same direction and 2 when opposite directions.

The optimization model used in the three network training is the Gradient Descent (GDS). The GDS optimization algorithm follows the gradient of the trainable parameters to minimize the loss function. The trainable parameters in our case are neuron weights and bias of both layers of each network. The GDS model weight update is defined as the following: -

Given a loss function $F(W)$

Then, the weights are updated in accordance with the following equation:

$$W_{ij} := W_{ij} - lr\frac{\partial}{\partial W_{ij}}F(W) \tag{11}$$

where

- lr: Learning rate (step size)

The main factors of the equation are the error derivative with the respect to weights and the learning rate. The error derivative with the respect to the weight is a common factor to, the loss function derivative, the activation function derivative, and former neurons outputs. The loss computed for the hidden layer in backpropagation is based on the chain rule for multi-variant function.

Algorithm 2 The Training Algorithm

Require: Query Features φ_I
Require: Target Feature φ_T
Require: Network $Model \leftarrow Net(\varphi_I, \varphi_T, hidden\_layer)$
Require: Loss Function $loss$
  if φ_I.shape > 1 **then**
    $\varphi_I \leftarrow$ concatenate(φ_I) ▷ For NetC
  end if
  while $loss < threshold$ **do**
    for `iteration` in `batch` **do**
      $OpNet \leftarrow Model(\varphi_I)$
      $loss \leftarrow loss(\varphi_T, OpNet)$
      Propagate errors backward ($input \leftarrow output$)
      Accumulate weights changes
      Apply weight changes to Model
    end for
  end while

TABLE 1. The inputs targets and the loss functions used in the training.

Net	Input	Target	Loss function
NetA	φ_{Qx}	φ_{Qt}	Batch-Mean Square Error
NetB	φ_{Tt}	φ_{Tx}	Batch- mean (1-Cosine Similarity)
NetC	$\varphi_{QMt}, \varphi_{Qt}$	φ_{Tt}	Batch-Mean Square Error

The well-known PyTorch library APIs are used to apply the GDS backpropagation training for the networks. The backpropagation algorithm includes defining batch size to save training time and computation power. The networks are generated, and trainable parameters are set to random, then the training starts. The training process includes two main cycles forward cycle and backward cycle. In the forward cycle, inputs are applied and propagated to network outputs. The loss function is applied using network output and the input-target output to compute loss. During the backward cycle errors propagated from output to input and the weight updates are computed per network weight. The weights update applies to the network every batch. The training algorithm summarizes the process of training.-

V. EXPERIMENTAL RESULTS

In our experimentation, the ResNet architectures (18,50,152) are included in the study. The proposed three models were trained using the 200K fashion dataset training dataset. The networks are also trained in two phases parallel and tuning.

In the parallel phase, the three networks were trained in parallel and on perfect data from the training dataset. The inputs, targets, and loss functions used in the training are summarized in Table (1).

The later numbers are the cardinalities of the NetA input vector and NetB output vector per The proposed Network. NetC, the compositional network trained on the perfect setup as of Table (1) until MSE 0.003 and used for the three models. NetA trained per The proposed model on the training dataset as of Table (1). NetB inputs in semantic formation are affected by the propagated errors generated by the former two network stages. To specify this effect, let us assume P_{AE}, P_{BE}, P_{CE} are

TABLE 2. The proposed net performance on the dataset.

	Training Dataset		
	NetA	NetB	NetC
ResNet 18	0.0201	0.1454	0.0157
ResNet 50	0.0159	0.1313	0.0128
ResNet 152	0.0150	0.1353	0.0124
	Test Dataset		
ResNet 18	0.052	0.1319	0.0443
ResNet 50	0.022	0.1168	0.0214
ResNet 152	0.027	0.1203	0.0248

the mapping error probabilities of the three networks. Then, the expected errors at the three networks outcomes will be:

$$\overline{P_{AE}} = P_{AE} \tag{12}$$

$$\overline{P_{CE}} = P_{AE} + P_{CE} - P_{AE} * P_{CE} \tag{13}$$

$$\overline{P_{BE}} = \overline{P_{CE}} + P_{BE} + P_{CE} - \overline{P_{CE}} * P_{BE} \tag{14}$$

This error propagation to NetB inputs mandates two stages of training. The first stage uses the training dataset as in Table (1). The networks were trained until the mean error was below 0.12. NetB performance after the stage of training in modal formation was found to be 0.27, 0.22, and 0.21. In the second phase, the network inputs are replaced by NetC-output. The training set continues at 0.16 then recall performance validation is activated every ten iterations using a sample of 1000 elements. The results MSE of the training performance on the training and testing datasets are summarized in Table (2). the recalling performance of the proposed Net in comparison with recent studies [25] and [47] presented in Table (3).

During the second training stage of NetB, the three Nets' training behavior was different. In the case of ResNet18, network learning was oscillating around the reported results without any significant progress in error reduction or improvement in the recall capabilities. ResNet152 and ResNet50 around the reported results were in a slowly progressive learning state. However, the ResNet50 was less in MSE, and the recall ability of ResNet152 was higher than ResNet50. The improved recall performance of the ResNet152-based Net even at higher mean square error apparently comes from the discriminative ability and the features extracted quality of that network. ResNet18-based Semantic Net performance was between the reported in [25] and [47]. This net was unable to cross over [47] which is likely due to the insufficient number of features extracted, 512.

Samples of the performance of the three nets recalling performance are in Fig. 7, Fig. 8 and Fig. 9. The Figures contain four typical recall queries per network. The query image is on left. The arrow is labeled with the modifier string. The arrow points to the first five recalled images. Boxed images are the targets of the query. The whole results, not just the presented samples, do not contain far-odd recalls. The odd recall meant here is recalling trousers when the query image is a jack for example. Looking in general at the presented samples we can easily see that all the recalls are not far from the target choices.

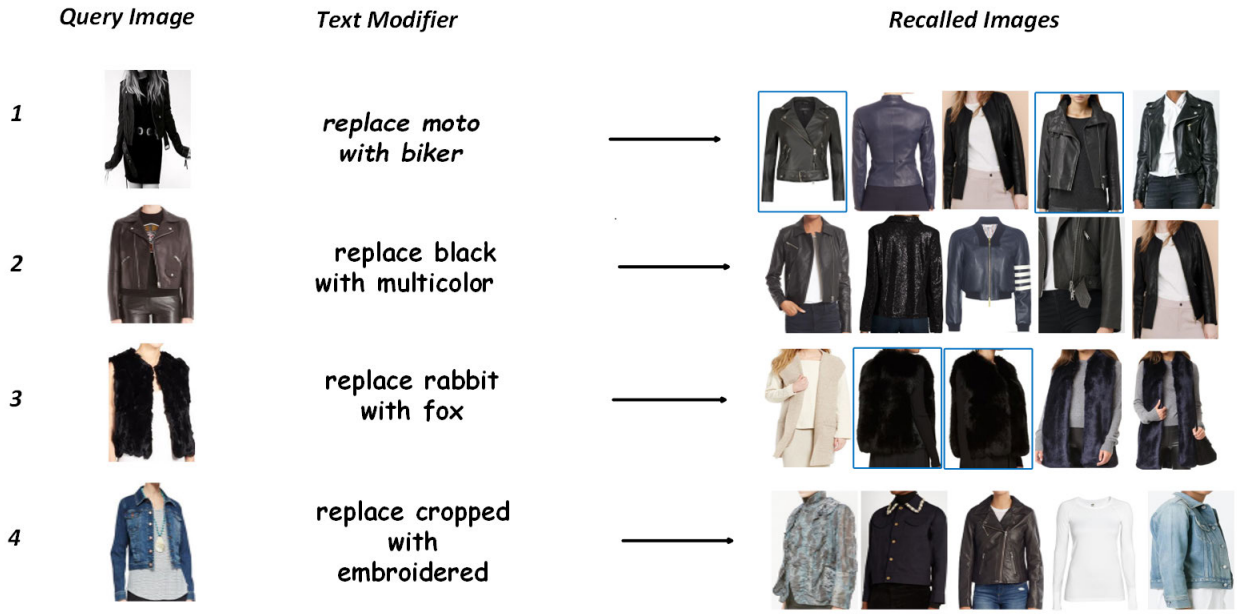


FIGURE 7. Typical recalling examples using the proposed architecture of ResNet-18 models.

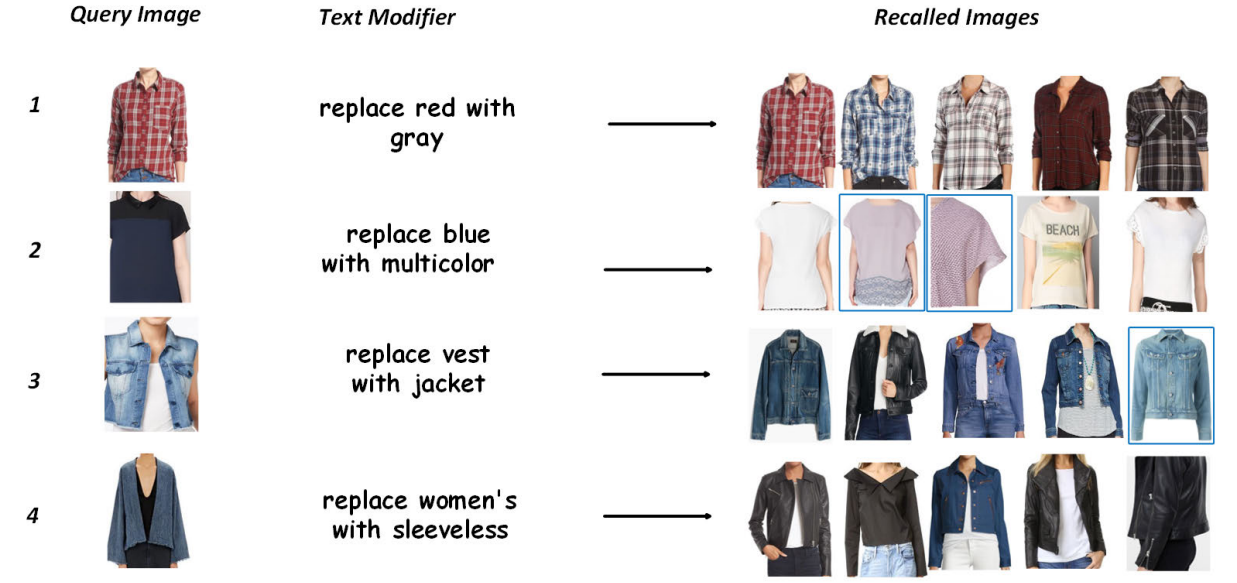


FIGURE 8. Typical recalling examples using the proposed architecture of ResNet-50 models.

In Net-18, Fig.7, the first presented recall has a hit on the first recalled image and even the fourth recall is a hit. In the second presented recall, one can easily infer that color change is not fully comprehended by the network as the first recall has the target to change color. The third case contains two hits, and the other two cases are very close. In the fourth recall, it seems that the presented image dress confused the network as it contains two pieces of different anatomies however all recalled images carry significant features related to the query.

In Net-50, Fig. 8, The first recall query indicates that the network comprehended the gray color and failed to some extent to fully integrate with the image features however the five images carry significant features from the recall image. The multicolor comprehension here is not clear if comes from the presented image or text comprehension. The second recall contains two hits and the other two are very close. The third recall of the network, the last image was a hit, and the rest are very close to target too. The last recall of the network points to the fact that the network failed to comprehend the changes to the design of the outfit.

In Net-152, Fig. 9, the results speak for themselves. In general, the whole results of the queries are close to the targets. However, there are a few odd cases such as two first images in the third case That point to the quality of the generated features by the net supersedes the other two networks.

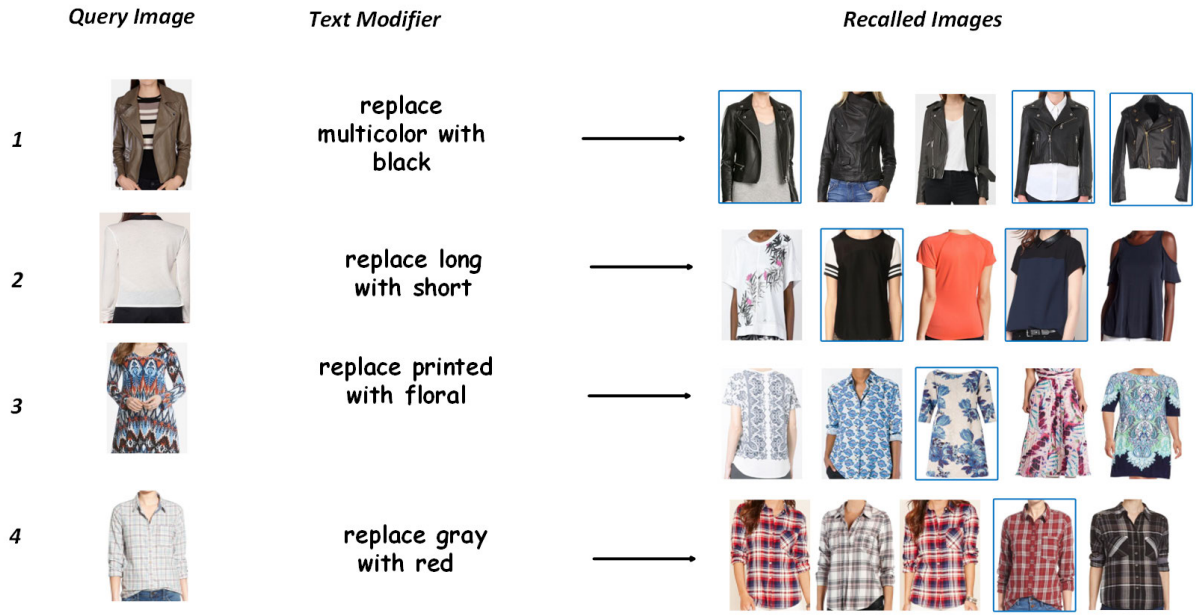


FIGURE 9. Typical recalling examples using the proposed architecture of ResNet-152 models.

TABLE 3. The proposed net results in comparison with [25] and [47] and others.

Method	R@1	R@10	R@50
Han et al	6.3	19.9	38.3
Image only	3.5	22.7	43.7
Text only	1.0	12.3	21.8
Concatenation	11.9±1.0	39.7±1.0	62.6±0.7
Show and Tell	12.3±1.1	40.2±1.7	61.8±0.9
Param Hashing	12.2±1.1	40.0±1.1	61.7±0.8
Relationship	13.0±0.6	40.5±0.7	62.4±0.6
MRN	13.4±0.4	40.0±0.8	61.9±0.6
FiLM	12.9±0.7	39.5±2.1	61.9±1.9
TIRG	14.1±0.6	42.5±0.7	63.8±0.8
TIRG with BERT	19.9±0.6	51.7±1.5	71.8±1.3
Compose AE	22.8±0.8	55.3±0.6	73.4±1.5
Net 18	19.86	46.12	67.54
Net 50	23.44	48.68	72.275
Net 152	25.4	52.85	74.51

The overall performance of the three networks' recall does not contain far too odd recall cases. In fact, the result indicates that the proposed architecture is viable. The overall network performance is as expected Net 50 better than Net18 and Net 152 better than Net 50. The shortcoming in the results from our point of view comes from the textual-semantic representations that are used in the embedding layer prior to the LSTM. This embedding technique is based on Continuous Bag Of Words (CBOW) and Skip Gram (SG) based on the assumption that similar words are used in a similar context. In our case, for example, all the colors are considered similar, and their embedded vectors will be of insignificant distance as they are used in a typical context. However, from a query point of view, they should be of significant distance. That interprets the query results, the second of Net 18 and the first of Net 50.

VI. CONCLUSION

In this paper, a proposed CBIR methodology with query inputs text and image is introduced. The provided text is a modifier that is intended to be applied to the input image to form a conceptual target image. The conceptual generated image will be used to recall similar images from the image database. The proposed methodology solution for this CBIR problem uses ResNets pre-trained models on image net for image features extraction and the LSTM neural network with word embedding for text features extraction. The fully connected layers of the two well-known deep networks, LSTM and ResNet are replaced jointly by a proposed Net. The proposed Net is composed of three interconnected single hidden fully connected feedforward interconnected neural networks. The proposed Net uses semantically viable architectures to compose features. The three non-linear feedforward single-hidden layer networks are called NetA, NetB, and NetC. The NetA's role is to comprehend textually the image. That is, produces image-caption features from the query image features. NetC performs feature composition. In other words, it applies the Modifier text features to query image text features to formulate possible target image caption features. NetB maps target-image caption features to target-image features. The formed target image features are used to recall images from the image-base based on the cosine similarity metric. The proposed Net mimics the human brain steps when similar queries are presented to him. Three architectures from the proposed Net are built using the most well-known ResNet architectures 18, 50, and 152. A total of seven single hidden layer networks are built 3-NetA, 3-NetB, and 1-NetC. The networks are trained in two phases on the well-known Fashion200K training dataset. In the first phase of the training, the six networks were parallel trained using

PyTorch on Google-Colab using dataset training data. In the second phase, the three NetB were finetuned to compensate for the cascade error propagation. The proposed models were tested on the Fashion200K testing dataset. The statistical results of the three networks' performance were found to be comparable with recent studies [25], [47]. The visual inspection of random query samples assured that the design is viable. However, there are some shortcomings in the recalls. We believe that it could be improved, and that will be the focus of our future work.

In future works, the proposed architecture could be improved in several dimensions which include network regularization, Dataset, and phased verifications. Network regularization helps in adapting the neural networks to learn the model rather than learning the training dataset itself. Network regularization during training could be done by using Weight Decay, Activation functions Decay, and variable learning rate. Moreover, network regularization could be done by changing the network architectures such as adding Dropout layers, adding Batch Normalization layers, and changing network Hyper-parameter. Changing network hyperparameters includes adding more hidden layers, adding more neurons, changing non-linear functions, and even using other optimistic optimization techniques. The dataset improvement dimension includes verification of dataset validity to network training to the intended function and embedding semantic quality inspection. In dataset training validity, as the network will not learn for example changing color from black to blue from a few cases, then comes the question which needs a specific answer ''Are there enough generated cases for the network to learn the intended function?''. In embedding: ''Are the generated word-embedded vectors generated by the embedding layer semantically viable to the problem? In the phased verification, as each network performs specific function mapping that could be verified a testing dataset could be generated for each network for verification and generalization assurance. A Net with verified components will assure great overall performance.

REFERENCES

[1] M. Aboali, I. Elmaddah, and H. E.-D. Hassan, ''Augmented TIRG for CBIR using combined text and image features,'' in *Proc. Int. Conf. Electr., Comput. Energy Technol. (ICECET)*, Dec. 2021, pp. 1–6, doi: 10.1109/ICECET52533.2021.9698617.

[2] E. R. Davies, *Computer Vision: Principles, Algorithms, Applications, Learning*, 5th ed. New York, NY, USA: Academic, 2018.

[3] N. Kondylidis, M. Tzelepi, and A. Tefas, ''Exploiting tf-idf in deep convolutional neural networks for content based image retrieval,'' *Multimedia Tools Appl.*, vol. 77, no. 23, pp. 30729–30748, Dec. 2018.

[4] Z. Zheng, L. Zheng, M. Garrett, Y. Yang, and Y.-D. Shen, ''Dual-path convolutional image-text embeddings with instance loss,'' *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 16, no. 2, pp. 1–23, 2017.

[5] J. Li, ''The application of CBIR-based system for the product in electronic retailing,'' in *Proc. IEEE 11th Int. Conf. Comput.-Aided Ind. Design Conceptual Design*, Yiwu, China, Nov. 2010, pp. 1327–1330.

[6] R. U. Khan, X. Zhang, R. Kumar, and E. O. Aboagye, ''Evaluating the performance of ResNet model based on image recognition,'' in *Proc. Int. Conf. Comput. Artif. Intell.*, 2018, pp. 86–90.

[7] G. Van Houdt, C. Mosquera, and G. Nápoles, ''A review on the long short-term memory model,'' *Artif. Intell. Rev.*, vol. 53, pp. 5929–5955, Dec. 2020.

[8] P. Ferré, F. Mamalet, and S. J. Thorpe, ''Unsupervised feature learning with winner-takes-all based STDP,'' *Frontiers Comput. Neurosci.*, vol. 12, p. 24, Apr. 2018.

[9] I. Misra, A. Gupta, and M. Hebert, ''From red wine to red tomato: Composition with context,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1116–1169.

[10] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, ''Neural module networks,'' in *Proc. CVPR*, 2016, pp. 39–48.

[11] D. D. Hoffman and W. A. Richards, ''Parts of recognition,'' *Cognition*, vol. 18, nos. 1–3, pp. 65–96, 1984.

[12] P. Felzenszwalb, D. McAllester, and D. Ramanan, ''A discriminatively trained, multiscale, deformable part model,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[13] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu, ''Image parsing: Unifying segmentation, detection, and recognition,'' *Int. J. Comput. Vis.*, vol. 63, no. 2, pp. 113–140, 2005.

[14] Z. Si and S.-C. Zhu, ''Learning AND-OR templates for object recognition and detection,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2189–2205, Sep. 2013.

[15] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, ''Learning to compose neural networks for question answering,'' in *Proc. NAACL*, 2016, pp. 1–10.

[16] B. M. Lake, R. Salakhutdinov, J. Gross, and J. B. Tenenbaum, ''One shot learning of simple visual concepts,'' in *Proc. 33rd Annu. Conf. Cognit. Sci. Soc.*, vol. 172, 2011, p. 2.

[17] V. Krishnan and D. Ramanan, ''Tinkering under the hood: Interactive zero-shot learning with net surgery,'' 2016, *arXiv:1612.04901*.

[18] M. Xin and Y. Wang, ''Research on image classification model based on deep convolution neural network,'' *EURASIP J. Image Video Process.*, vol. 2019, pp. 1–11, Dec. 2019.

[19] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, ''Multimodal compact bilinear pooling for visual question answering and visual grounding,'' 2016, *arXiv:1606.01847*.

[20] H. Noh, P. H. Seo, and B. Han, ''Image question answering using convolutional neural network with dynamic parameter prediction,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 30–38.

[21] D. Teney, P. Anderson, X. He, and A. Van Den Hengel, ''Tips and tricks for visual question answering: Learnings from the 2017 challenge,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4223–4232.

[22] A. Miech, I. Laptev, and J. Sivic, ''Learnable pooling with context gating for video classification,'' 2017, *arXiv:1706.06905*.

[23] J. Chen, Y. Shen, J. Gao, J. Liu, and X. Liu, ''Language-based image editing with recurrent attentive models,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8721–8729.

[24] X. Han, Z. Wu, P. X. Huang, X. Zhang, M. Zhu, Y. Li, Y. Zhao, and L. S. Davis, ''Automatic spatially-aware fashion concept discovery,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1463–1471.

[25] N. Vo, L. Jiang, C. Sun, K. Murphy, L.-J. Li, L. Fei-Fei, and J. Hays, ''Composing text and image for image retrieval—An empirical Odyssey,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019.

[26] T. Nagarajan and K. Grauman, ''Attributes as operators: Factorizing unseen attribute-object compositions,'' in *Proc. CVF Conf. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018.

[27] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, ''Show and tell: A neural image caption generator,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3156–3164.

[28] R. C. Staudemeyer and E. R. Morris, ''Understanding LSTM—A tutorial into long short-term memory recurrent neural networks,'' 2019, *arXiv:1909.09586*.

[29] F. Malik and B. Baharudin, ''Analysis of distance metrics in content-based image retrieval using statistical quantized histogram texture features in the DCT domain Author links open overlay panel,'' *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 25, no. 2, pp. 207–218, 2013.

[30] S. R. Dubey, ''A decade survey of content based image retrieval using deep learning,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, pp. 2687–2704, May 2022.

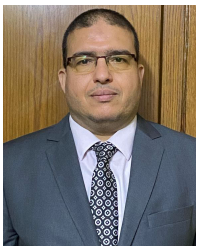
[31] T. Jaworska, ''Query techniques for CBIR,'' in *Flexible Query Answering Systems 2015* (Advances in Intelligent Systems and Computing), vol. 400. Springer, 2016.

[32] X. Wang, K. Liu, and X. Tang, ''Query-specific visual semantic spaces for web image re-ranking,'' in *Proc. CVPR*, 2011, pp. 1–8.

[33] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, "The QBIC project: Querying images by content using colour," in *Proc. SPIE*, vol. 1908, 1993, pp. 173–187.
[34] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, Nov. 2000.
[35] V. E. Ogle and M. Stonebraker, "CHABOT: Retrieval from a relational database of images," *Computer*, vol. 28, no. 9, pp. 40–48, Sep. 1995.
[36] X. Shi, M. Sapkota, F. Xing, F. Liu, L. Cui, and L. Yang, "Pairwise based deep ranking hashing for histopathology image classification and retrieval," *Pattern Recognit.*, vol. 81, pp. 14–22, Sep. 2018.
[37] J.-H. Lim and J. S. Jin, "A structured learning framework for content-based image indexing and visual query," *Multimedia Syst.*, vol. 10, no. 4, pp. 317–331, Apr. 2005.
[38] N. Rasiwasia, P. J. Moreno, and N. Vasconcelos, "Bridging the gap: Query by semantic example," *IEEE Trans. Multimedia*, vol. 9, no. 5, pp. 923–938, Aug. 2007.
[39] J. Jing, T. Gao, W. Zhang, Y. Gao, and C. Sun, "Image feature information extraction for interest point detection: A comprehensive review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4694–4712, Apr. 2021.
[40] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2425–2433.
[41] X. Guo, H. Wu, Y. Cheng, S. Rennie, G. Tesauro, and R. S. Feris, "Dialog-based interactive image retrieval," 2018, *arXiv:1805.00145*.
[42] H. Nam, J.-W. Ha, and J. Kim, "Dual attention networks for multimodal reasoning and matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 299–307.
[43] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. Van Den Hengel, "Visual question answering: A survey of methods and datasets," *Comput. Vis. Image Understand.*, vol. 163, pp. 21–40, Oct. 2017.
[44] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, "A simple neural network module for relational reasoning," in *Proc. NIPS*, 2017, pp. 1–10.
[45] J.-H. Kim, S.-W. Lee, D. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang, "Multimodal residual learning for visual QA," in *Proc. NIPS*, 2016, pp. 1–9.
[46] Y. Zhang and H. Lu, "Deep cross-modal projection learning for image-text matching," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 686–701.
[47] M. U. Anwaar, E. Labintcev, and M. Kleinsteuber, "Compositional learning of image-text query for image retrieval," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 1139–1148.
[48] *The Fashion200K Dataset*. Accessed: Sep. 2020. [Online]. Available: https://github.com/xthan/fashion-200k
[49] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 2, no. 1, pp. 1–19, Feb. 2006.
[50] A. R. Lahitani, A. E. Permanasari, and N. A. Setiawan, "Cosine similarity to determine similarity measure: Study case in online essay assessment," in *Proc. 4th Int. Conf. Cyber IT Service Manage.*, Apr. 2016.
[51] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, "Euclidean distance geometry and applications," *SIAM Rev.*, vol. 56, no. 1, pp. 3–69, 2012.
[52] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 201–216.
[53] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1096–1104.
[54] T. Nagarajan and K. Grauman, "Attributes as operators: Factorizing unseen attribute-object compositions," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 169–185.
[55] E. Perez, F. Strub, H. D. Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–10.
[56] F. Tan, P. Cascante-Bonilla, X. Guo, H. Wu, S. Feng, and V. Ordonez, "Drill-down: Interactive retrieval of complex scenes using natural language queries," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 2647–2657.
[57] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5005–5013.

MOHAMED A. ABOALI received the B.Sc. degree from the Computer Science and Engineering Department, Modern Academy, Cairo, Egypt, in 2012. He is currently a Software Engineer with German University in Cairo (GUC). His research interests include deep learning, artificial intelligence, and image processing.

ISLAM ELMADDAH received the M.Sc. degree in computer engineering from Ain Shams University, Egypt, in 1999, and the Ph.D. degree from King's College London, U.K., in 2004. He is an Assistant Professor with the Department of Computer Engineering, Faculty of Engineering, Egypt. He teaches courses related to software engineering and requirements analysis. His current research interests include data mining, the Internet of Things, intelligent instrumentation, and software engineering.



HOSSAM E. ABDELMUNIM received the B.Sc. and M.S. degrees in electrical engineering from Ain Shams University, Cairo, Egypt, in 1995 and 2000, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Louisville, Louisville, KY, USA. He joined the Computer Vision and Image Processing Laboratory (CVIP Laboratory), University of Louisville, in June 2002, where he has been involved in the applications of image processing and computer vision for medical image analysis. He is currently a Full Professor with the Computer and Systems Engineering Department, Faculty of Engineering, Ain Shams University. He has authored or coauthored more than 75 technical articles, including ICIP, MICCAI, ICCV, CVPR, BMVC, DICTA, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and *Journal of Real-Time Image Processing* (JRTIP). His current research interests include image modeling, image segmentation, and deep learning.