




Review

# A Survey on Symmetrical Neural Network Architectures and Applications

Olga Ilina <sup>1</sup>, Vadim Ziyadinov <sup>1</sup>, Nikolay Klenov <sup>1,2,\*</sup> and Maxim Tereshonok <sup>1,3</sup>

<sup>1</sup> Science and Research Department, Moscow Technical University of Communications and Informatics, 111024 Moscow, Russia; o.v.ilina@mtuci.ru (O.I.); v.v.ziyadinov@mtuci.ru (V.Z.); m.v.tereshonok@mtuci.ru (M.T.)

<sup>2</sup> Faculty of Physics, Lomonosov Moscow State University, 119991 Moscow, Russia

<sup>3</sup> Moscow Institute of Physics and Technology, 141700 Dolgoprudny, Russia

\* Correspondence: nvklenov@gmail.com

**Abstract:** A number of modern techniques for neural network training and recognition enhancement are based on their structures' symmetry. Such approaches demonstrate impressive results, both for recognition practice, and for understanding of data transformation processes in various feature spaces. This survey examines symmetrical neural network architectures—Siamese and triplet. Among a wide range of tasks having various mathematical formulation areas, especially effective applications of symmetrical neural network architectures are revealed. We systematize and compare different architectures of symmetrical neural networks, identify genetic relationships between significant studies of different authors' groups, and discuss opportunities to improve the element base of such neural networks. Our survey builds bridges between a large number of isolated studies with significant practical results in the considered area of knowledge, so that the presented survey acquires additional relevance.

**Keywords:** symmetrical neural network; Siamese neural network; triplet neural network; neural network structure; neural network training; signature verification; speech verification; face analysis; semantic text analysis; image retrieval; re-identification; stereo matching; visual object tracking; change detection



**Citation:** Ilina, O.; Ziyadinov, V.; Klenov, N.; Tereshonok, M. A Survey on Symmetrical Neural Network Architectures and Applications. *Symmetry* **2022**, *14*, 1391. <https://doi.org/10.3390/sym14071391>

Academic Editor: Mihai Postolache

Received: 6 June 2022

Accepted: 3 July 2022

Published: 6 July 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The applications of artificial neural networks (ANNs) are widespread because of their generalizing ability that allows finding similarities and differences between objects. Modern ANNs can find non-obvious complex features for objects and build generalizing rules. It allows use of ANNs for recognition, object-to-object comparison, verification, etc.

Researchers have used symmetry in recent years in the world of artificial neural networks, giving impressive results on various object recognition accuracies using special architectures. These architectures allow the informative features of these objects to be distinguish. Among different ANN architectures used for comparison and verification, researchers distinguish Siamese neural networks (SNN). SNNs are symmetrical jointed structures that take two or more objects as input and allow determination of their similarity measures in a certain feature space, which varies depending on the subject area characteristics.

An important feature of Siamese structures is the “merging” of identical ANNs with each other using the same array of weight coefficients obtained during the pre-training process. The other feature of Siamese networks—symmetry—is used for a parallel translation of initial descriptions of compared objects into a feature space convenient for their comparison. Since Siamese networks learn not to recognize objects, but to form their features suitable for comparison (and similarity/difference estimation), these networks do not require additional training when new data are entered. Siamese networks used for different

tasks differ in the specifics of feature space forming, in which objects are compared, and the calculation of the similarity/difference measure.

The rest of the paper is organized as follows. Related works are shown in Section 2. The main idea of Siamese and triplet architectures and the history of their first implementations are summarized in Section 3. In Section 4, we describe the history of Siamese and triplet architectures' development for such applications as signature verification, speech verification, face analysis, semantic text analysis, image retrieval, re-identification, stereo matching, visual object tracking, and change detection—each in its own subsection. We provide a review of state-of-the-art implementations, training methods and summarize the architectures in schemes. In Section 5 we discuss the implementations of large symmetrical neural networks on the hardware level—the main directions of hardware neural networks development. The discussion of survey results and application trends is provided in Section 6.

## 2. Related Works

As of 2022, several analytical reviews involving Siamese and triplet neural networks [1–4] have been published. In [1], a wide range of works (164 references) on the Siamese neural networks' application in different fields of science and technology are mentioned. The analysis of architectures and working principles is not conducted. The author only notes works and their main topics. In contrast, paper [2] contains a deep and detailed review of symmetrical neural network approaches for a limited range of visual object tracking tasks. The comparison of various tracking algorithms involving Siamese neural networks is presented. This area of knowledge is actively developing and still contains many significant unsolved problems. In the review, ref. [3], the general principles of Siamese and triplet neural networks used in computer vision, as well as their performance on the selected datasets, are considered. The main advantages of triplet networks are shown to simplify data preprocessing (normalization and calibration). In [4] the application of different metrics (loss functions) in Siamese and triplet neural network training for image recognition and re-identification is considered. At the same time, the consideration of different Siamese and triplet network architectures is limited.

This survey closes the above-described gaps in the systematics of the symmetrical neural networks and aims to present the reader with a generalized analysis of modern architectures and a wide variety of Siamese neural network applications, as well as trends in their development, not being limited to the computer vision tasks. Thus, this survey combines both breadth of coverage and depth of source analysis, as shown in Table 1.

**Table 1.** Comparison of related surveys.

Surveys	Various Applications	Various Architectures	Hardware Implementations
Chicco, D. [1]	+	-	-
Ondrašovič, M.; Tarábek, P. [2]	-	+	-
Nandy, A. et al. [3]	-	+	-
This survey	+	+	+

## 3. General Architecture Background and History of Symmetrical Neural Networks

The first example of a symmetrical neural network architecture with common weights for two identical modules is given in [5]. Bromley et al. described a neural network for two fingerprints similarity estimation. The architecture presented is Siamese in the classical definition, but the term “Siamese” was first used in 1994 in [6], where the problem of finding forged handwritten signatures was considered (the verified signature and signature in memory were processed in identical subnetworks to obtain feature vectors, and then the distance between two vectors was calculated). Bromley et al. [6] claimed a verification accuracy of 95.5%.

The next mention of SNNs occurred in 2005 with a face verification task [7], where convolutional neural networks were used as symmetrical Siamese modules. For its training, the concept of “learning similarity” was introduced and a contrastive loss function was proposed. Further, the development of symmetrical neural network architectures and the expansion of their application list grew.

Traditionally, the task of an SNN is to train the function  $f_w$  to map a pair of input data  $x_1$  and  $x_2$  into a smaller dimension space so that the distance  $d$  between feature vectors  $f_w(x_1)$  and  $f_w(x_2)$  is small for semantically similar input data  $x_1$  and  $x_2$ , and large for input data having different classes [7]. A simplified structure of the described Siamese architecture is shown in Figure 1.

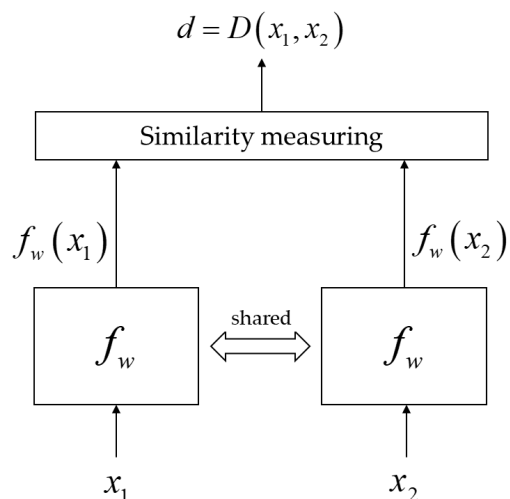


Figure 1. Simplified Siamese neural network scheme.

Various ANN architectures are used as the mapping function  $f_w$ . The contrast function is traditionally used for their training [7]:

$$L(d, y) = (1 - y) \cdot L_G(d) + y \cdot L_I(d)$$

where  $d = D(x_1, x_2)$  is the distance metric between input data maps  $f_w(x_1)$  and  $f_w(x_2)$ ;  $y$  is a binary label with  $y = 0$  for similar input data (genuine pair) and  $y = 1$  otherwise (impostor pair);  $L_G(\cdot)$  is monotonically increasing loss function for a genuine pair;  $L_I(\cdot)$  is monotonically decreasing loss function for an impostor pair.

The authors in [7] proposed the next loss function:

$$L(d, y) = (1 - y) \cdot \frac{2}{Q} \cdot (E_w)^2 + y \cdot 2Q \cdot e^{-\frac{2.77}{Q} E_w},$$

where

$$E_w(x_1, x_2) = ||f_w(x_1) - f_w(x_2)||$$

is the energy function to measure the similarity of input data  $x_1$  and  $x_2$ ,  $Q$  is set to the upper bound of the energy function.

Another contrastive loss formula follows:

$$L(d, y) = (1 - y) \cdot d^2 + y \cdot \max(0, m - d)^2$$

where  $m$  is the margin (the typical value is 1).

Manhattan metric and Euclidean metric are often used as distance metrics  $d(a, b)$  between different vectors  $a$  and  $b$ .

Along with Siamese architecture, triplet networks (consisting of three identical neural network architectures with common weights) also exist and are actively developed [8].

Such architectures take three data samples as inputs: anchor  $x$ , positive sample  $x^+$  (the same class as anchor  $x$ ) and negative sample  $x^-$  (a different class from anchor  $x$ ). In this case, the output of the neural network consists of two distance metrics between mappings of input pairs  $(x, x^+)$  and  $(x, x^-)$ , obtained after ANN  $f_w$ . A simplified triplet network architecture is shown in Figure 2.

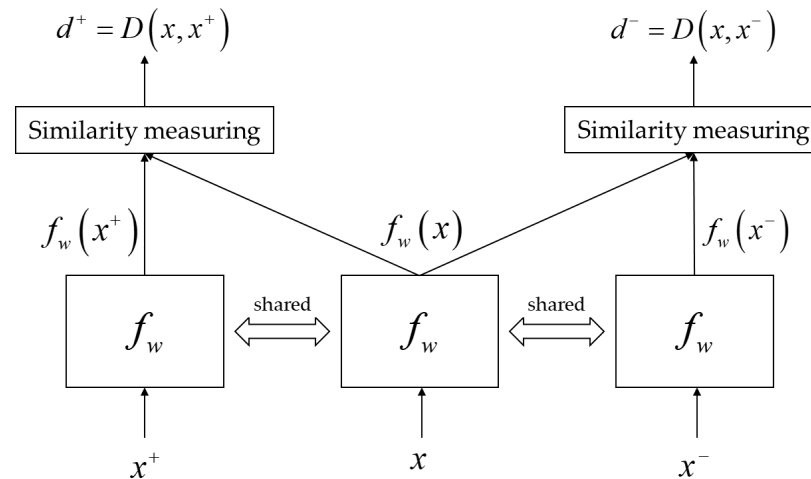


Figure 2. Simplified triplet network scheme.

The triplet network learns to decrease the semantic distance  $d^+$  between the anchor input and the positive sample and increase the distance  $d^-$  between the anchor input and the negative sample simultaneously.

#### 4. Contemporary Applications of Siamese Architectures

To review and analyze applications with Siamese neural network architectures, we have searched Scopus, IEEEExplore, Google Scholar, and Web of Science databases and selected the most frequently cited publications. The choice criterion was the active citation of the selected paper and the number of papers concerning the selected generalized theme. Over a hundred papers were found, and during the consequent analysis we have chosen more than 80 most influencing works and studied them thoroughly. During the Siamese networks' application analysis, we have revealed the nine most popular directions united in four big groups (Figure 3). Further in this paper, we describe each of these applications in detail (Sections 4.1–4.9), review the most significant publications, and analyze and visualize the evolution of SNN usage methods.

In these subsections, we thoroughly consider the system and subnetwork architectures, methods of input data preprocessing, input data dimensionality transformation, method of similarity measure calculation, and loss functions used for training. To describe the subnetwork architectures of Siamese structures in the following sections, we apply the following layer designations (Table 2).

In case the subnet architecture does not have a settled name, it is represented as a chain of designations shown in the Table 2.

##### 4.1. Signature Verification

Handwritten signature verification is the easiest and most common way for person identification in banking, insurance, forensics, etc. Despite its simplicity and variety of uses, however, handwritten signatures require a system for verification. Traditional methods are based on hidden Markov models, dynamic matching, and rule-based algorithms. Siamese neural networks, considering their symmetrical architecture, are also well suited for signature verification [9–13]. Figure 4 illustrates the chronology of scientific papers dedicated to Siamese neural network applications in signature verification tasks; the isolation of up to 40% of significant works in this direction is clearly seen.

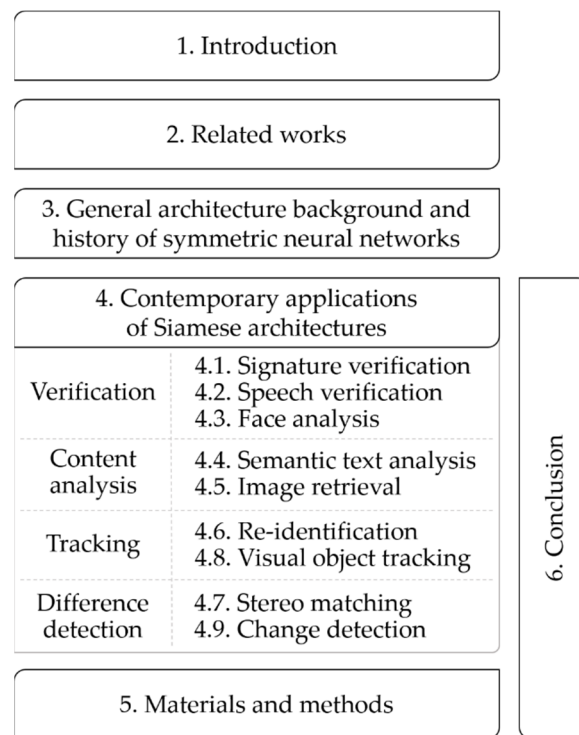
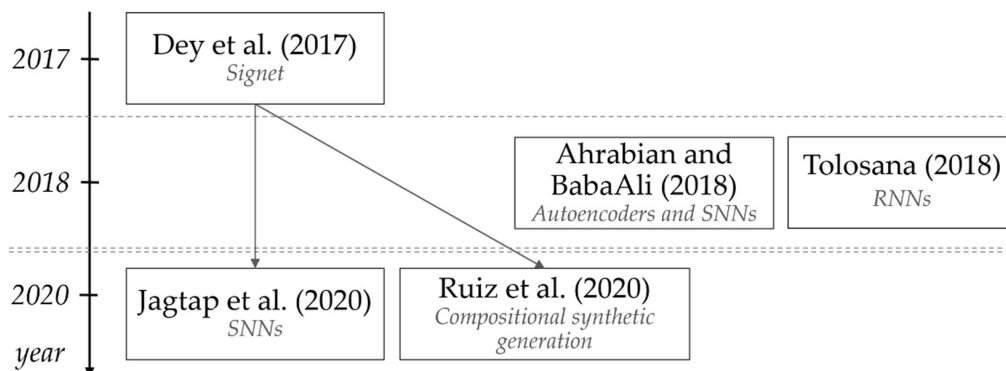


Figure 3. Article structure and basic applications of symmetrical neural networks.

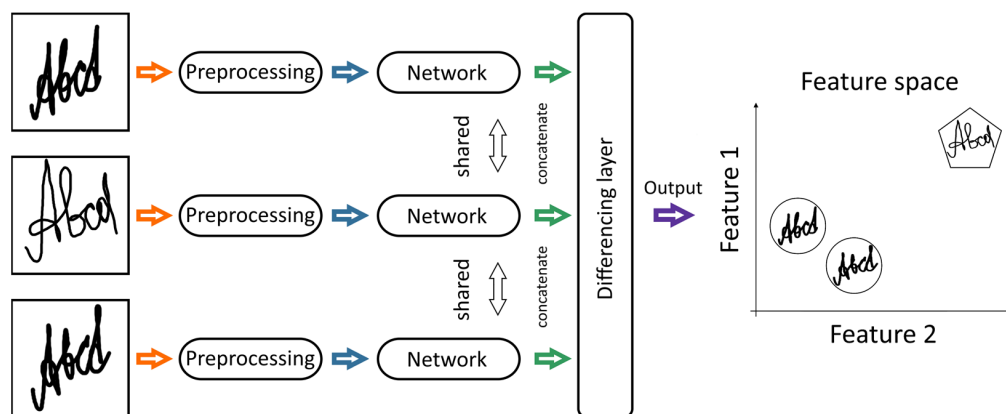
Table 2. Layer designations in neural networks.

Designation	Description
@k1×k2×f	Convolutional layer with kernel size k1 × k2, output filters number f
@k1×k2	Max pooling layer with kernel size k1 × k2
@k1×k2	Average pooling layer with kernel size k1 × k2
@f	Fully connected layer, number of output neurons f
@f	Recurrent neural network (RNN), output space dimension f
@f	Bi-directional RNN, output space dimension f
@f	Long short-term memory (LSTM) layer, output space dimension f
@f	Bi-directional LSTM layer, output space dimension f
@f	Gated recurrent units (GRU) layer, output space dimension f
@f	Bi-directional GRU layer, output space dimension f

Signature verification systems can be categorized by the data source: static (offline) or dynamic (online) systems. Static systems take the two-dimensional image after the signature acquisition process as input. Dynamic systems take information about the signing process. It means the signature is represented as a sequence of signal values received by a detection device. Online signature representation is more informative for verification. Also, there are writer-independent systems that do not require modification when new signers are added, and writer-dependent systems that require re-training when new signers are added (it is an important disadvantage). A general view of the network architecture is shown in Figure 5.



**Figure 4.** Timeline and links for major works on the Siamese neural networks in the signature verification task. The following works are mentioned: Ahrabian and BabaAli (2018) [9]; Tolosana (2018) [10]; Dey et al. (2017) [11]; Jagtap et al. (2020) [12]; Ruiz et al. (2020) [13]. The summaries of these works are provided in Section 4.1.



**Figure 5.** Simplified SNN structure for signature verification. The input to the system is an image of the original signature (top), as well as a variant of the forged signature (middle) and an image of the original signature. The Siamese neural network acts as a function to approximate the authentic signature to the original and to distance the forged signature in the feature space.

Ahrabian et al. [9] propose an architecture with an autoencoder to translate the received data into a fixed-length hidden space and SNN to process extracted samples. The paper also uses the attention mechanism to get a model capable of focusing on the most important features of the input data, which are the 12 local features extracted from the pen sequences. The proposed architecture, besides improving verification accuracy, requires less computational cost than traditional methods based on dynamic time warping (DTW).

Currently, recurrent neural networks (RNN) are actively used for online signature verification. In [10] the SNN architecture based on four RNN variants is considered: long short-term memory (LSTM), gated recurrent unit (GRU), and their bi-directional variants (BLSTM and BGRU). The paper presents the first successful scheme of multiple RNN systems usage (i.e., LSTM and GRU) for online handwritten signature verification. The architectures described in this paper receive 23 time functions extracted from compared signatures. These functions include 12 local features, as well as some additional features. Tolosana et al. [10] highlight the excellent learning capability of the proposed architectures even for a few signatures.

In offline handwritten signature verification, convolutional neural networks (CNNs) are used as part of SNNs. CNN usually takes a two-dimensional image of a signature as input. The paper, Ref [11], describes a framework with writer-independent feature extraction using SigNet architecture that takes a pair of images—with a genuine signature and a signature for verification. The architecture returns embedding vectors for each

input image to compute semantic distance [14]. In [11] a comparison of the proposed SigNet architecture with non-neural network methods for signature verification is given. It turned out that the architectures based on neural networks outperform the most advanced competitors on most of the datasets. Also, these systems demonstrate improved fraud detection accuracy for different signers and forgers. Jagtap et al. [12] optimized the SigNet architecture [11] by extending embedding vectors and obtained an SNN with an even better ability to recognize forgeries and verify genuine signatures. Further, Ruiz et al. [13] proposed an SNN using the Inception module in CNN. The proposed architecture has fewer training parameters but achieves higher accuracy in recognizing forged signatures and is suitable for use by new signers without an additional training.

Figure 4 shows that publications describing methods of online signature verification using Siamese architectures [9,10] are of little interest at present, in contrast to SNN-based offline signature verification methods [11–13]. Figure 6 illustrates the parameters of the signature verification methods discussed in this section for more detailed information.

Method	Input data preprocessing	Input data sizes	Simplified diagram or architecture name of SNN branch	Similarity measuring/ Unifying module	Loss function
Ahrabian and BabaAli (2018)	Standard Normalization, Crop length	12 local features			Binary cross-entropy
Tolosana (2018)		23 time functions			
Dey et al. (2017)	Resize, Invert image, Normalize	155×220		Euclidean distance	Contrastive loss
Jagtap et al. (2020)				Euclidean distance	Contrastive loss
Ruiz et al. (2020)	Binarization, Morphological erosion, Resize, Normalization	128×128		Manhattan distance	Binary cross-entropy

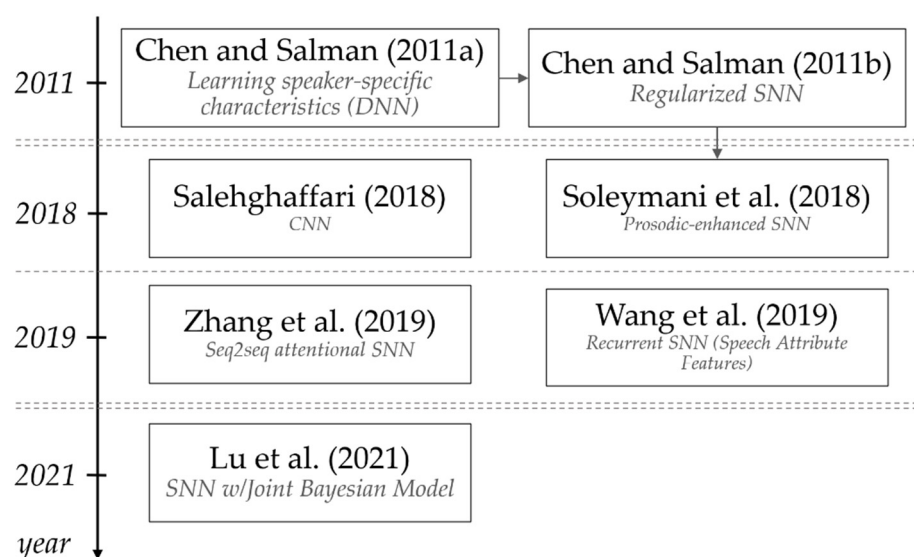
**Figure 6.** Schematic representation of input data preprocessing algorithms, sizes, SNN branch architectures, algorithms for similarity measurement (or merging modules), and loss functions for each signature verification method. The designations of the architecture layers are presented in Table 2. The following works are mentioned: Ahrabian and BabaAli (2018) [9]; Tolosana (2018) [10]; Dey et al. (2017) [11]; Jagtap et al. (2020) [12]; Ruiz et al. (2020) [13].

To estimate the performance of signature verification methods, metrics are evaluated on a variety of datasets. Therefore, it is difficult to evaluate the quality of signature verification and to compare it with other approaches and systems. However, as shown in [15], existing SNNs demonstrate superiority over most competitors.

#### 4.2. Speech Verification

Speech verification (speaker verification)—methods of determination if the speaker matches the declared one. Speech verification methods are based on the comparison of some input data and pre-recorded information. It is used in security systems, forensics, information security, and, more recently, as a method of biometric user authentication. Speech conveys a large amount of mixed information—both semantic and speaker-specific spectral. Extracting components necessary for verification is non-trivial.

New algorithms resistant to various interferences (external environment noise, various characteristics of sound recording devices, interference of telephone channels, etc.) have been recently developed in this field. Also, the problems of attacks on speech verification systems are being commonly solved. There are several datasets for speech verification, including YOHO, VoxCeleb, RSR2015, and others. Figure 7 illustrates the timeline and links for major scientific papers dedicated to Siamese neural network applications in speech verification tasks.



**Figure 7.** Timeline and links for major works in speech verification. The following works are mentioned: Chen and Salman (2011a) [16]; Chen and Salman (2011b) [17]; Lu et al. (2021) [18]; Soleymani et al. (2018) [19]; Salehghaffari (2018) [20]; Zhang et al. (2019) [21]; Wang et al. (2019) [22]. The summaries of these works are provided in Section 4.2.

A distinction is made between text-dependent (using fixed phrases) and text-independent verification methods. Most works are concerned with text-independent verification methods due to their wider applicability.

Early SNN architectures in speaker verification appeared in 2011 [16,17]. The approaches proposed in these works include multi-objective loss functions to amplify the influence of typical speaker-specific features, insignificant information clearance, and hybrid deep neural network learning. Hybrid learning consists of preliminary layer-by-layer (local) unsupervised learning with a greedy algorithm, and then global supervised learning. The authors demonstrated that the system reliably verifies speakers, regardless of whether spoken phrases were used in training, and regardless of the language spoken. Comparative studies have shown that this approach demonstrates superiority over other available systems for speaker verification.

Lu et al. [18] considered the speaker verification as a Bayesian binary classification task and proposed a Siamese neural network architecture with a new approach—training with “matching” and “different” pairs of samples. Using the recorded speech parameter matrices, a logarithmic likelihood estimation is computed, which is then converted into the distance metric. Further, the distance metric is used for training. Linear transformation



functions are implemented as separate layers of the Siamese neural network. The authors proved the advantages of the SNN learning algorithm for the speaker verification task.

Soleymani et al. [19] proposed a new text-independent Siamese convolutional neural network architecture for speaker verification suitable for various devices. The proposed architecture uses Mel-frequency spectrogram coefficients instead of the popular approach using Mel-frequency cepstral coefficients, acquiring the ability to analyze spectral and temporal feature dependencies. Although the use of spectral-temporal features has shown high reliability in speaker verification models, the system uses only speech in a short time interval. Authors proposed an improved Siamese convolutional neural network architecture, using a multilayer perceptron (MLP) to include prosodic, jitter, and shimmer features. The proposed end-to-end verification architecture performs feature extraction and verification simultaneously. The proposed architecture demonstrates significant progress over classical and deep algorithms.

Salehghaffari et al. [20] used the following approach: a convolutional neural network is trained to distinguish different speakers' identities to create a background model. Then SNN is used to create a feature space. The novelty of the approach is the fine-tuning of the trained neural network model. It is claimed that this approach has robustness to variations in speaker parameters, both prosodic and linguistic, as well as sound recorder and environment parameters.

The sequence-to-sequence attentional Siamese neural network (Seq2Seq-ASNN) was introduced in [21]. This method moved away from the pronunciation-based speaker parameter estimation approach to the temporal matching information methodology. The proposed model concentrates on the frames of the sequence (Seq2Seq), maps each frame estimation representation into a feature space, and generates an estimation vector based on the pronunciation parameters to calculate the final measure of similarity. Experimental results demonstrate the superiority of the proposed model over various basic methods, including traditional i-Vector/PLDA, end-to-end speaker verification models, d-vector approaches, and a self-attention model for text-dependent speaker verification on the Tencent dataset.

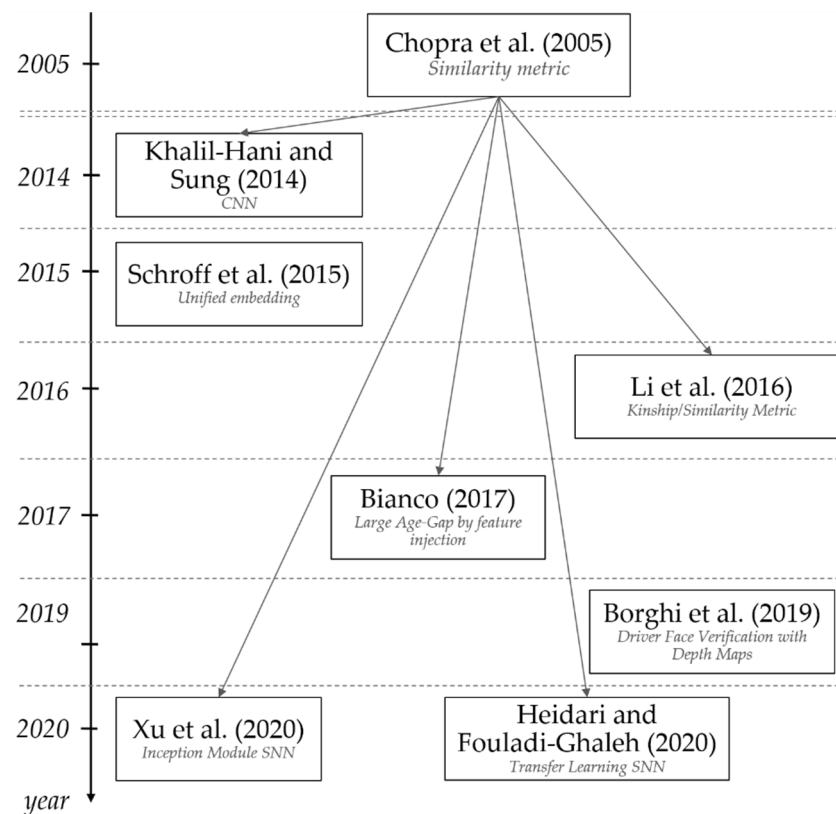
A new system design that requires only speech recordings for training is proposed in [22]. This system is used for the detection of children's speech disorders. The system focuses on fast embedding capability and is based on a recurrent Siamese neural network architecture, which is trained by computing similarity and divergence parameters of pronunciation between two audio recordings. Previously, similar approaches required a large dataset of recordings with speech disorders, which was a significant problem. To detect speech disorders, the trained network computes a sequence of variable length feature vectors. Then it measures the distance between the vectors and the reference vectors obtained during the training. The model also includes a binary classifier. The results of the experiments conducted by Wang et al. show that a recurrent Siamese network using a combination of the reference and spoken speech features can achieve a detection accuracy of 0.941.

It follows from the analysis that the development of the deep neural networks application made speaker verification methods more accessible. The proposed methods provide high accuracy, but Siamese neural networks in the tasks of speech verification have not yet found wide popularity among the authors.

### 4.3. Face Verification

Face verification (or authentication) aims to decide whether two input face images belong to the same person [23].

Figure 8 illustrates a timeline of face verification method development. As one can see from the figure below, the most popular methods were mainly based on the work of [7] (references are indicated by arrows).



**Figure 8.** Timeline and links for major works in face verification methods. The following works are mentioned: Chopra et al. (2005) [7]; Khalil-Hani and Sung (2014) [24]; Schroff et al. (2015) [25]; Li et al. (2016) [26]; Bianco (2017) [27]; Borghi et al. (2019) [28]; Xu et al. (2020) [29]; Heidari and Fouladi-Ghaleh (2020) [30]. The summaries of these works are provided in Section 4.3.

Khalil-Hani et al. [24] continued the development of [7] and propose to use an SNN comprising two CNNs for the face verification task, where each CNN is reduced to four layers by merging convolution layers and sub-sampling. Network training is performed using the annealing global learning rate method. The authors show that the proposed CNN architecture can classify a pair of face images significantly faster compared with the equivalent network architecture with a cascade of convolution and sub-sampling layers [7].

Schroff et al. [25] have developed the FaceNet system for face recognition, verification, and clustering tasks, which comprises a triplet network. Such a network is fed with triplets consisting of anchor input and positive and negative samples. The authors introduce triplet loss to train such architecture:

$$L = \sum_{n=1}^N \left( (d^+)^2 - (d^-)^2 + \alpha \right)$$

where  $d^+ = \|f_w(x) - f_w(x^+)\|$  is a semantic distance between the anchor input and the positive sample;  $d^- = \|f_w(x) - f_w(x^-)\|$  is a semantic distance between the anchor input and the negative sample;  $\alpha$ , margin; and  $N$ , the number of triplets in the dataset. This training method allows direct mapping into a compact Euclidean space, where distances directly correspond to the facial similarity measure. In [25] it is shown that using a 128-dimensional feature vector extracted from a face image by deep CNNs, a high efficiency of face recognition, verification, and clustering is achieved.

The similarity metric based convolutional neural network (SMCNN) solving kinship verification tasks was first proposed in [26]. This network takes as input two images of close relatives' faces (for example, father–son, father–daughter, mother–son, mother–daughter). Images are run through CNN, then the feature vectors are extracted to measure the semantic

distance between the samples. Li et al. [26] compared the quality of kinship verification for the proposed method and state-of-the-art algorithms that do not use Siamese structures. It turned out that SMCNN architecture achieves the best results for almost all kinship verification. The exception is mother–son relations, for which the best results were obtained using the deep neural network with hand-crafted features.

In [27] a method for face verification with large age differences is presented, as well as a large age-gap (LAG) dataset with images ranging from child/young to adult/old. The proposed method includes a pre-trained deep convolutional neural network (DCNN) for the face recognition task on the large dataset. Further, it was fine-tuned on the LAG dataset using Siamese architecture. A feature injection layer is introduced to improve verification accuracy. Feature injection was used in the first DCNN layer to combine externally computed features with activations of the deepest DCNN layers. Experimental results on the LAG dataset show that the proposed method can outperform competing algorithms.

Borghi et al. [28] proposed a driver face verification method involving depth maps. The proposed architecture is designed for automotive applications where it is necessary to obtain data even in the absence of light sources, which led to the use of depth maps calculated from time-of-flight (ToF) sensors (i.e., near-infrared sensors). Instead of the contrast loss function from [7], the authors proposed to use concatenated feature maps from the Siamese architecture to get one feature map. This feature map is fed to two additional convolutional layers, with the latter sigmoidal activation function. The output provides a number between 0 and 1 to estimate the similarity measure of the two input faces. Borghi et al. [28] claim that the proposed architecture shows acceptable accuracy, working in real time even on CPUs and embedded boards.

In [29] it was proposed to use the Inception module incorporated Siamese convolutional neural network (IMISCNN) and the cyclical learning rate policy to train it. The IMISCNN architecture comprises two CNNs with a single Inception module containing more information due to different receptive fields. Cyclical learning rate (CLR) is introduced to optimize the learning rate of the proposed IMISCNN. The proposed learning method and neural network architecture provide fast convergence in the learning process and demonstrate a higher face verification accuracy compared with other architectures, including the original SNN [7].

Heidari et al. [30] proposed Siamese networks with transfer learning using deep CNN [31] on small datasets. The obtained results indicate an increase in the accuracy compared with other similar methods involving training on small datasets.

Figure 9 illustrates the parameters of the face verification methods considered in this section, namely: information on input data pre-processing, input data size, information on the architectures found in each Siamese structure branch, input data similarity metrics, and loss functions.

Figure 9 shows that the concept of the Siamese structure has changed little since the original. The exception is the work of [28], where a small neural network is used as a unifying module being a part of the architecture. Also, a different loss function is used in [28]. Schroff et al. [25] suggested using a triplet network architecture. A new loss function was developed for this architecture.

Also, from Figure 9, one can conclude that there is a trend of increasing complexity of SNN architectures with increasing input data and reducing the number of image pre-processing operations, indicating a gradual transition to automatic face verification methods. However, there are few methods for face comparison using the similarity learning concept. Researchers are concerned with face identification, i.e., the person's determination on the input image.

Method	Input data preprocessing	Input data sizes	Simplified diagram or architecture name of SNN branch	Similarity measuring / Unifying module	Loss function
Chopra et al. (2005)	simple correlation-based centering algorithm	56×46		Energy function	Contrastive loss
Khalil-Hani, Sung (2014)	normalization to get pixels from 0 to 1	46×46		Manhattan distance	Logistic loss
Li et al. (2016)		64×64			
Bianco (2017)	face alignment algorithm	200×200	pre-trained AlexNet 		Contrastive loss
Schroff et al. (2015)	centering	from 96×96 to 244×244		Euclidean distance	Triplet loss
Heidari, Fouladi-Ghaleh (2020)		128×128	pre-trained VGG-16 		Contrastive loss
Xu et al. (2020)		72×72		Energy function	
Borghini et al. (2019)	face detection	100×100 depth maps			Binary cross-entropy

**Figure 9.** Schematic representation of input data preprocessing algorithms, input data sizes, SNN branch architectures, algorithms for similarity measurement (or merging modules), and loss functions for each method. Designations of architecture layers are presented in Table 2. The following works are mentioned: Chopra et al. (2005) [7]; Khalil-Hani and Sung (2014) [24]; Schroff et al. (2015) [25]; Li et al. (2016) [26]; Bianco (2017) [27]; Borghini et al. (2019) [28]; Xu et al. (2020) [29]; Heidari and Fouladi-Ghaleh (2020) [30].

#### 4.4. Semantic Text Analysis

Semantic similarity search is a natural language processing (NLP) task that shows a quantitative similarity measure between texts or documents using a specific metric. Semantic similarity detection methods are used to search for spam, to assess the relevance of a topic to a statement, to compare the content of two documents (in anti-plagiarism systems), etc. The chronology of the Siamese semantic text analysis methods development is illustrated in Figure 10.

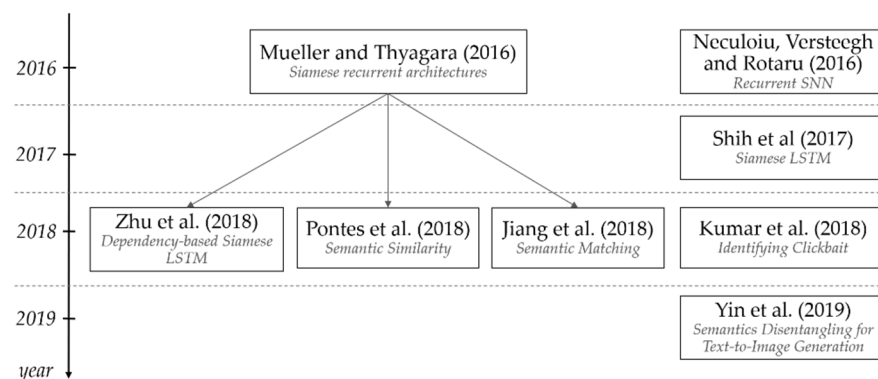
Mueller et al. [32] presented a Siamese version of the long short-term memory (LSTM) network for estimating semantic similarity between sentences based on the Manhattan metric. This system maps a pair of variable-length input sentences into a pair of fixed-length vectors to compute the distance between them, which reflects a similarity measure.

The paper, ref. [33], presents a deep architecture for learning the similarity of variable-length sentences. The authors propose using bidirectional LSTMs (BLSTMs) as part of a Siamese architecture. It uses similarity information between pairs of variable-length strings to map them into a fixed-dimension space. It has been shown in several experiments that when the training dataset is extended with typos, synonyms, and extra words, the model shows invariance to such distortions.

Also, Siamese architecture based on LSTM has found its application in categorization and text classification, information retrieval, spam detection, etc. [34]. Semantic representation of documents is obtained after the document has passed through an LSTM encoder, which maps word embeddings into a document representation fixed-length vector. The paper by Shih et al. used pre-trained word embeddings. The proposed architecture categorizes the input text with high accuracy.

In [35] yet another Siamese architecture is proposed. It combines the CNN and LSTM architectures. To analyze the local context of words in a sentence and to create an idea of the word relevance and its neighborhood, Pontes et al. proposed the convolutional layer as part of the Siamese architecture. An LSTM layer application is necessary to analyze the whole sentence based on its words and local context. The authors argue that, because of the local context of the words, the analysis of the overall context increases accuracy.

Zhu et al. [36] proposed the Dependency-based Siamese LSTM network (D-LSTM). It aims to determine the similarity of the two sentences. Each Siamese architecture branch, in this case, contains a basic and a supporting component (BC and SC, respectively). LSTM is used to train the basic component, and Stanford parser [37] is used to train the supporting component. To get the final sentence mapping, the BC and SC outputs pass through a weighted summer. Then, a weighting factor is introduced to correct the importance of the main and auxiliary components. The D-LSTM architecture developed by authors attenuates the influence of adjectives or negotiable words in the input sentences, allowing stronger sentence representations to be examined for better similarity detection.



**Figure 10.** Timeline and links for major works in semantic text analysis methods. The following works are mentioned: Mueller and Thyagara (2016) [32]; Neculoiu, Versteegh and Rotaru (2016) [33]; Shih et al (2017) [34]; Pontes et al. (2018) [35]; Zhu et al. (2018) [36]; Kumar et al. (2018) [38]; Jiang et al. (2018) [39]; Yin et al. (2019) [40]. The summaries of these works are provided in Section 4.4.

In [38] a multi-strategy approach is developed to detect internet clickbait. The proposed model comprises a bi-directional LSTM (BLSTM) branch with an attention mechanism. It comprises two SNNs to determine the header effect on the clickbait evaluation. One SNN is used to find the similarity between the post title and its target description. Another SNN evaluates the relevance of the image attached to the post. The output of these three described components is combined and fed into a fully connected layer. The result is a probability that the post, along with its associated information, can be considered a clickbait.

In [39] a Siamese multi-depth attention based hierarchical recurrent neural network (SMASH RNN) is proposed. It performs semantic text matching of long documents. According to the Siamese structure, the proposed architecture has two multi-depth attention-based hierarchical RNN (MASH RNN). For each document, MASH RNN creates an informative representation based on the knowledge of different levels of the document structure (paragraph, sentences, and words). To create a multi-level representation of input information, the Siamese architecture contains an attention-based hierarchical RNN to encode each text level. The encoders outputs are combined to produce the document's final representation.

Yin et al. [40] developed semantics disentangling generative adversarial networks (SD-GAN) to synthesize photorealistic images based on the input text description. The structure of the proposed system is an SNN, each branch of which comprises a serially connected text encoder and a hierarchical generative adversarial network (GAN). A bi-directional long short-term memory (BLSTM) is used as a text encoder, which extracts semantic vectors from the input text description. The semantic vectors obtained from the encoder output go through hierarchical stages, with the generator at each stage. These generators aim to upsample images from low to high resolution to create a photorealistic image. Also, the authors introduced the semantic-conditioned batch normalization (SCBN) module, which focuses on identifying semantic commonalities. SCBB also ignores unique semantic differences in the input text to enhance the visual-semantic embedding in the feature maps of the generative networks. The experiments conducted by Yin et al. demonstrate the high efficiency of the proposed architecture for the CUB dataset and the complex large-scale MS-COCO dataset. The paper does not describe a technique to measure semantic text similarity. Yet, the proposed method uses information about the similarity or difference of two input texts to train a GAN sequence to generate images with specific content.

Figure 11 demonstrates the methods described above. For semantic text analysis methods using a Siamese structure, we can observe a rapid development from single-layer models with LSTM to complex, diverse multilevel models. The recurrent models' applications, in combination with many modern neural network solutions, breaks new ground in semantic text analysis.

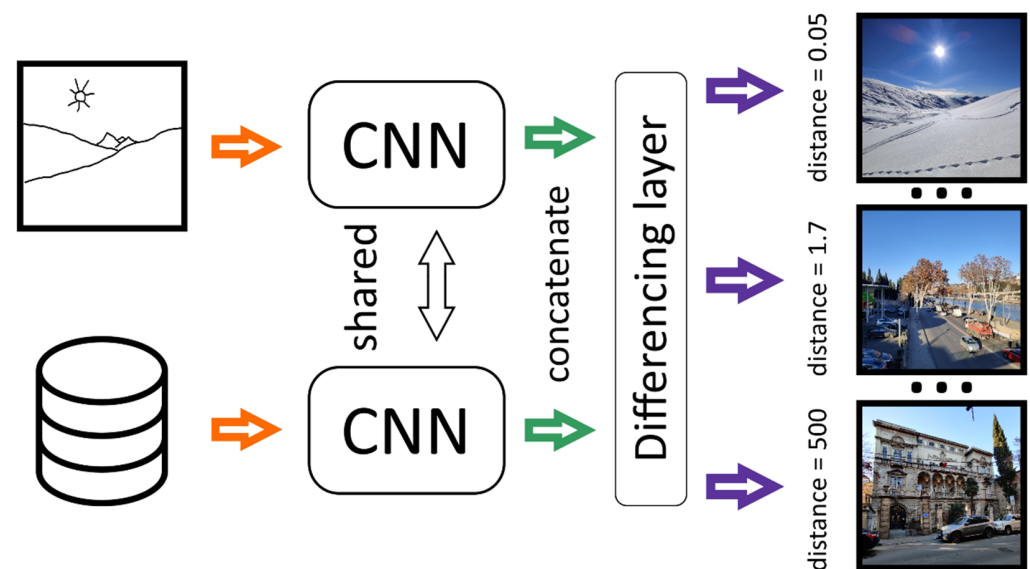
Method	Input data preprocessing	Input data sizes	Simplified diagram or architecture name of SNN branch	Similarity measuring / Unifying module	Loss function
Mueller & Thyagara (2016)	word2vec	Variable		Manhattan distance	MSE
Pontes et al. (2018)					
Zhu et al. (2018)					
Neculoiu, Versteegh and Rotaru (2016)	Padding to produce a sequence with 100 characters	Variable		Cosine distance	Contrastive loss
Shih et. al (2017)	word2vec or Glove vector representation			Euclidean distance	
Yin et al. (2019)					
Kumar et al. (2018)	Post Text → 300 - dimensional Doc2Vec Post Title → 300 - dimensional Doc2Vec Post Image → VGG-19 + FC@300	224×224		Text Embedding → Element-wise product Visual Embedding → Element-wise product Concatenate → @32	Binary cross-entropy loss

**Figure 11.** Schematic representation for methods of semantic analysis of textual information. Designations of architecture layers are presented in Table 2. The following works are mentioned: Mueller and Thyagara (2016) [32]; Neculoiu, Versteegh and Rotaru (2016) [33]; Shih et al (2017) [34]; Pontes et al. (2018) [35]; Zhu et al. (2018) [36]; Kumar et al. (2018) [38]; Yin et al. (2019) [40].

#### 4.5. Image Retrieval

Image retrieval systems search and retrieve the most similar images to a given input. The similarity parameters between the input image and the images from the database are used to rank them in order of similarity. Image retrieval methods are used in various applications. For example, they are applied to search for images by corresponding sketches, to search for clothes in a store by photo, etc.

For most databases, image retrieval by keywords or annotations is a difficult task because of the ever-increasing image databases. Touch-screen devices allow retrieval of imaginative, concise, and abstract finger-drawn images (sketches) (Figure 12). So, using sketches as a request is attractive since it is uncomplicated. It has led to the development of various methods of automatic sketch-based image retrieval (SBIR). Figure 12 illustrates how SNN-based SBIR systems work.



**Figure 12.** Schematic presentation of the image extraction systems principle. By feeding a sketch to the SNN input, the output will be a series of images ranked by the semantic distance between the sketch and the photos retrieved from the database.

Wang et al. [41] proposed a sketch-based 3D shape retrieval model based on CNN. It allows retrieval of 3D models from 2D hand sketches. To solve the problem of cross-domain learning, the authors suggested extending the basic version of CNN to SNN (one branch for projections of 3D models, the other for sketches); CNNs are trained separately.

The Siamese convolutional neural network (SCNN) for image retrieval, where a hand sketch is used as a query, is considered in [42]. Qi et al. proposed to use the Euclidean distance as a similarity measure and contrast loss function. The experiments have demonstrated that using SCNN for sketch-based image retrieval provides better performance compared with other methods.

In [43] a method for compact descriptor analysis of sketch-based image retrieval was proposed. For this task, Bui et al. proposed triplet ranking CNN and triplet loss. Three images are brought to input: query image and positive and negative samples. The authors have also shown that the combination of principal component analysis (PCA) and feature quantization methods can reduce the size of the image vector representation by 98% (from 3200 bits to 56 bits) with almost no loss of accuracy. Therefore, the method is suitable for deploying on mobile devices.

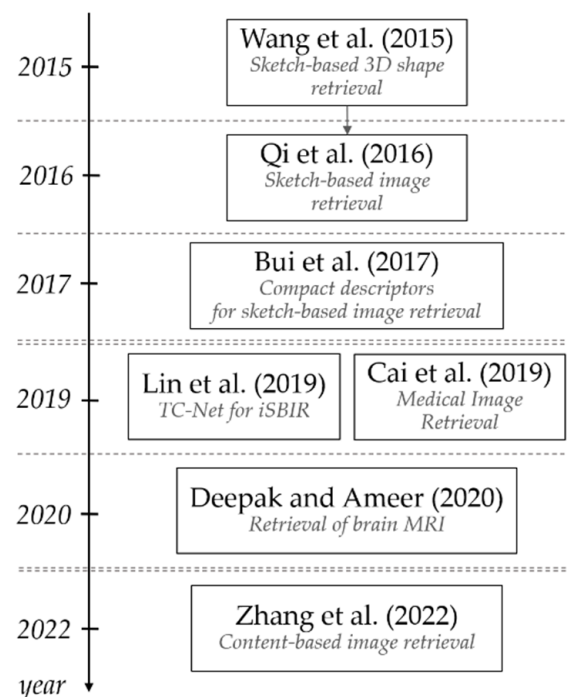
Lin et al. [44] developed the triplet classification network (TC-Net) for instance-level sketch-based image retrieval (iSBIR). The proposed triplet Siamese network includes a CNN and uses the global loss function, which is calculated as the sum of the triplet network loss and the auxiliary classification loss. Global function serves to minimize the distance

between the feature vectors extracted from the positive image and the anchor sketch. The experiments demonstrate the great capability of the proposed TC-Net for feature extraction for both thumbnails and images, which shows the classification loss function efficiency for the iSBIR problem.

In [45] a “medical image retrieval based on CNN and supervised” is proposed. Cai et al. propose to use SNN, each branch of which contains CNNs for feature extraction followed by hash mapping. Hash mapping is used to reduce the feature vectors dimensionality. Also, a new loss function with regularization term is introduced. The network architecture calculates a binary hash code from the input image. Subsequently, the Hamming distance between the obtained hash code and the hash codes of the images in the database is calculated. Thus, similar images sorted in order of Hamming distance are extracted from the database. The authors claim that the proposed method allows retrieval of images from the database faster and with higher accuracy than traditional hashing methods and some deep learning methods do.

A subset of image retrieval systems for analyzing image content (e.g., color, shape, texture, or other information extracted from an image) is called content-based image retrieval (CBIR). CBIR systems using SNN have found a wide application in medicine. For example, in [46] it was proposed to use a modified GoogleNet and transfer learning technology in an SNN structure for the magnetic resonance imaging (MRI) retrieval of the brain with tumor. Using a convolutional neural network in the structure of an SNN to distinguish lung cancer and tuberculosis on computed tomography (CT) images is described in [47]. Using modern CNNs as part of the SNN makes it possible to solve this complex diagnostic problem with higher accuracy.

Figure 13 illustrates the development chronology of image retrieval methods using Siamese architecture. One can see that research on the Siamese image extraction systems is still of interest. Schematic representation of image extraction methods is presented in Figure 14.



**Figure 13.** Timeline and links for major works in image extraction methods. The following works are mentioned: Wang et al. (2015) [41]; Qi et al. (2016) [42]; Bui et al. (2017) [43]; Lin et al. (2019) [44]; Cai et al. (2019) [45]; Deepak and Ameer (2020) [46]; Zhang et al. (2022) [47]. The summaries of these works are provided in Section 4.5.



Method	Input data preprocessing	Input data sizes	Simplified diagram or architecture name of SNN branch	Similarity measuring / Unifying module	Loss function
Wang et al. (2015)	Edge extraction	100×100		Manhattan distance	Contrastive loss
Qi et al. (2016)				Euclidean distance	
Bui et al. (2017)		Resize (save aspect ratio)	The longest side is 256		
Lin et al. (2019)	Convert sketches to 3 channels	225×225	pre-trained <i>DenseNet-169</i>	Spherical distance	Softmax Loss + Spherical Loss + Center Loss
Cai et al. (2019)		512×512		Hamming distances	Contrastive loss
Deepak and Ameer (2020)	Resize, normalize intensity values, convert to 3 channels	224×224		Euclidean distance	
Zhang et al. (2022)		70×70	pre-trained <i>SE-ResNet-101</i>		

**Figure 14.** Schematic representation of preprocessing algorithms, input data size, SNN branch architectures, algorithms for similarity measurement (or merging modules), and loss functions for each described image extraction method. The following works are mentioned: Wang et al. (2015) [41]; Qi et al. (2016) [42]; Bui et al. (2017) [43]; Lin et al. (2019) [44]; Cai et al. (2019) [45]; Deepak and Ameer (2020) [46]; Zhang et al. (2022) [47]. The designations of the architecture layers are presented in Table 2.

#### 4.6. Re-Identification

The re-identification term means techniques for matching anonymized data with known information. Most often, these are external data and the subjects of these data (identity verification). Siamese neural networks can identify and match identities in a video stream, photos, and photo combined portraits. Also, they are used to identify the luggage owner, as well as for vehicle re-identification and animal re-identification. These methods are useful for solving the object tracking task when some parameters of these objects or video streams change (different cameras, viewing angles, object appearance, etc.). Significant differences in a person's appearance because of different postures, viewpoints, changes in lighting, and various video stream distortions make human re-identification a difficult task. The increase in video recording quality, the number of cameras, and the computational processors' performance make this task urgent. Most of the considered

works offer a unique approach to Siamese neural network usage, which demonstrates the wide opportunities for re-identification task solutions.

Siamese neural networks for re-identification were first applied in [48]. Yi et al. described the convolutional neural network application, architecture, training process, and performed experiments to confirm the proposed method's superiority. To calculate the distance metric, the authors used the cosine distance layer.

McLaughlin et al. [49] proposed an architecture of Siamese neural network for re-identification with features extracted from each frame by a recurrent convolutional neural network. Comparison of the proposed architecture with the methods that do not use an SNN demonstrate a significant advantage for SNNs.

Siamese neural network architecture with long-term memory (LSTM) for re-identification is proposed in the paper, ref. [50]. The new architecture sequentially processes image areas and increases the local features' discriminatory ability using contextual information. Also, the LSTM detects spatial dependencies and selectively distributes relevant contextual information across the network.

Zheng et al. [51] propose a new Siamese deep learning neural network architecture that solves spatial localization and object appearance dependence. The proposed method includes the object's spatial localization into the learning process and combines the principles of Siamese learning and attention consistency. The authors have developed and described a consistent attentive Siamese network (CASN) architecture, which processes each individual image attentively.

Work on luggage and animal re-identification is also available. Using appearance as a feature in the luggage processing system allows processing of luggage more reliably and faster. For example, ref. [52] describes a model based on the Siamese neural network, capable of identifying luggage by its image.

The ability to re-identify animals is also of great importance for a wide range of tasks in ecology and biology, such as studying population dynamics or ecosystem functioning. Schneider et al. [53] demonstrate that triplet networks provide superiority over Siamese ones for all considered animal species. Also, the new technique demonstrated the superiority over the human level.

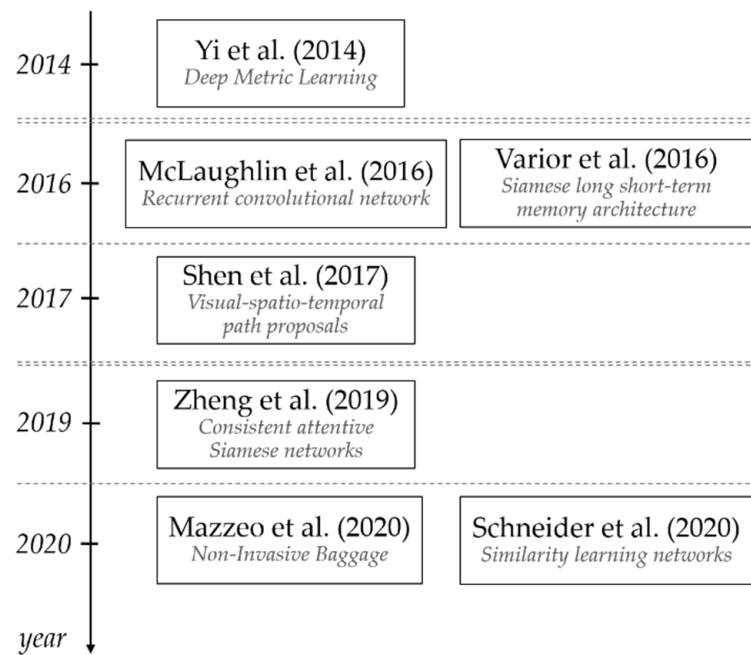
Vehicle re-identification plays an important role in intelligent traffic surveillance systems. The paper, ref. [54], proposed the Siamese-CNN + PathLSTM model, which is a two-stage system that uses both visual and spatiotemporal information. Also, Shen et al. stated the superiority of the proposed approach over modern methods using the VeRi776 dataset.

Figure 15 illustrates the development chronology of re-identification methods using Siamese architecture. Schematic representation of re-identification methods is presented in Figure 16.

Finally, we can conclude that the re-identification tasks are paid much attention and new techniques and architectures are being developed.

#### 4.7. Stereo Matching

Stereo matching is a depth map estimation technique (building a three-dimensional model) using several images taken from different points. Depth estimation and object distance on images are the basis for solving localization and tracking problems and are used in, for example, autopilot systems. The images from the left and right cameras may be fed to the neural network input. So, the objects on the images are differently positioned in the horizontal plane (Figure 17). Hence, objects located closer to the camera are displaced farther (have greater disparity) than distant objects. Correspondingly, higher disparity values indicate lower depth values (Figure 18). The results of stereo matching are usually represented as depth maps. A depth map is a matrix containing information about the distances from the objects' surfaces to the camera.



**Figure 15.** Timeline and links for major works in re-identification methods. The following works are mentioned: Yi et al. (2014) [48]; McLaughlin et al. (2016) [49]; Varior et al. (2016) [50]; Zheng et al. (2019) [51]; Mazzeo et al. (2020) [52]; Schneider et al. (2020) [53]; Shen et al. (2017) [54]. The summaries of these works are provided in Section 4.6.

Method	Input data preprocessing	Input data sizes	Simplified diagram or architecture name of SNN branch	Similarity measuring / Unifying module	Loss function
Yi et al. (2014)		40×128		Cosine distance	Deviance loss
McLaughlin et al. (2016)	convert to the YUV color space, normalize, optical flow estimation	sequence length of 192 frames		Euclidean distance	Contrastive loss
Varior et al. (2016)	divide the input image into several horizontal stripes				Cross-entropy loss
Zheng et al. (2019)	Resize	288×144			MSE
Mazzeo et al. (2020)		250×250	pre-trained EfficientNet-B5		Binary cross-entropy loss
Schneider et al. (2020)		224×244	pre-trained DenseNet201, InceptionV3 or ResNet152	Euclidean distance	Contrastive loss

**Figure 16.** Schematic representation of preprocessing algorithms, input data size, SNN branch architectures, algorithms for similarity measurement (or merging modules), and loss functions for described re-identification methods. The designations of the architecture layers are presented in Table 2. The following works are mentioned: Yi et al. (2014) [48]; McLaughlin et al. (2016) [49]; Varior et al. (2016) [50]; Zheng et al. (2019) [51]; Mazzeo et al. (2020) [52]; Schneider et al. (2020) [53].

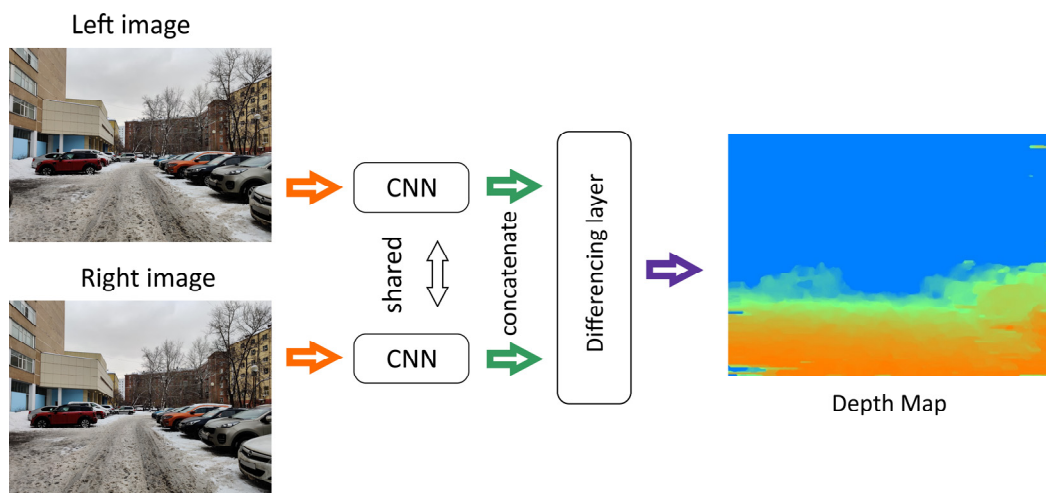


Figure 17. SNN depth map computation scheme.

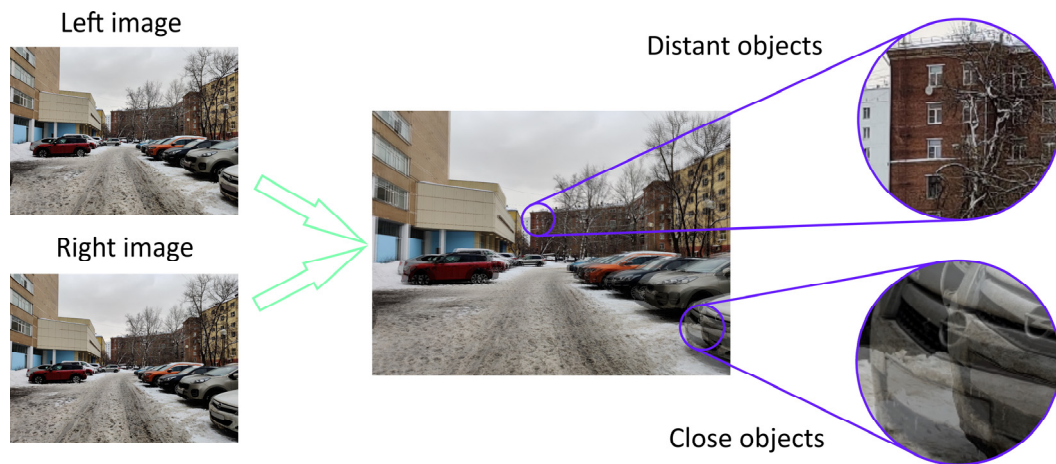


Figure 18. Example of stereo vision disparity.

Triangulation methods, epipolar geometry methods, or mathematical depth map computation can be used when obtaining data from a stereo system. Neural networks were first proposed to solve stereo matching problems in 2002 [55], and most of subsequent works were based on convolutional neural networks.

A Siamese neural network for stereo matching was first applied in [56] in 2014. The proposed system used the matching cost calculation by learning the similarity measure using a convolutional neural network. The convolutional neural network output is used to calculate the stereo matching distance.

Zbontar and LeCun [57] extended the research and described two network architectures—“fast” and “accurate”. Siamese neural networks are applied in both architectures and comprise several convolution and ReLU layers in the subnetworks and also the merging parts. The subnetworks calculate vectors reflecting the input image properties. The resulting vectors are compared using a cosine similarity coefficient to obtain the network’s result. The authors compared the results of the described architectures and demonstrated their significant advantage in the depth estimation accuracy and running time on KITTI 2012, KITTI 2015, and Middlebury datasets. The “fast” architecture calculates depth maps 90-times faster than the “accurate” architecture, with a negligible increase in the error rate. Thus, convolutional neural networks are applicable for high-precision real-time depth map computation.

Zagoruyko and Komodakis [58] compare the architectures and confirm the significant advantage of Siamese neural networks. The Siamese neural network using a spatial pyramid pooling (SPP) layer is also described [59].

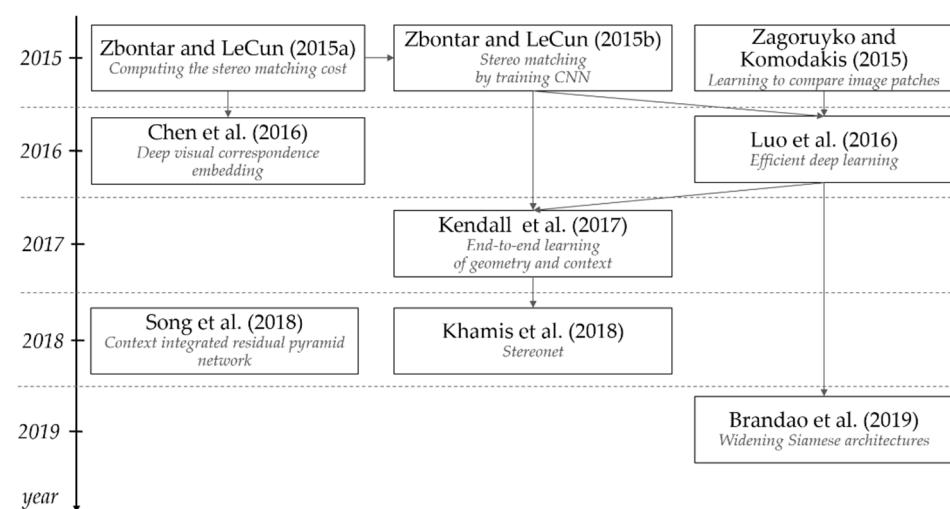
Luo et al. [60] have proposed a matching network that gives more accurate results than the MC-CNN-fst architecture [57] having comparable computational cost. This performance was achieved by replacing the entire block following Siamese subnets with a single layer performing scalar product. The task was considered as a classification; optimal smoothing techniques were proposed that also accelerated the network performance. Also, this paper contains an investigation of the probability distribution for all disparity values. The matching network proposed by the authors can produce accurate results in less than a second of computation using an up-to-date GPU. Similar solutions were implemented in [61].

In [62], the StereoNet, the first end-to-end deep architecture for calculating the object depth in real-time images, is presented. Khamis et al. state the use of hierarchical refinement to increase the dimensionality with edge preservation. Tests conducted by the authors show a nearly seven-fold reduction in incorrectly computed pixels compared with CG-Net Fast. Also, more than a two-fold reduction compared with CG-Net Full [63] is demonstrated on the SceneFlow dataset. Additionally, >4500 less processing time than MC-CNN-acrt and >50 less processing time than MC-CNN-fst [57] is achieved.

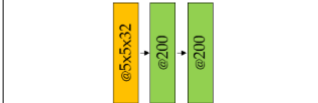
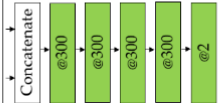
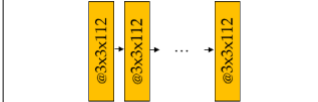
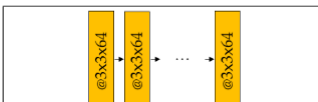
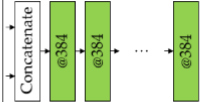
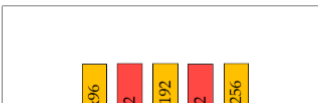
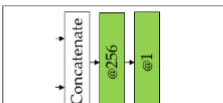
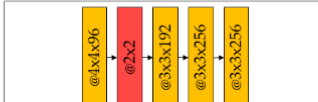

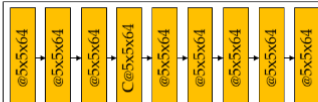
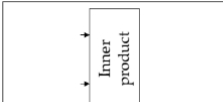
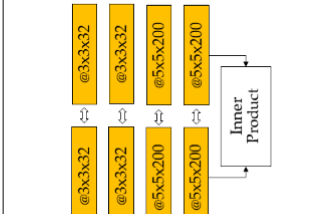
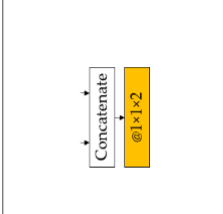
In [64] a multitasking architecture that uses an auxiliary edge subnet is presented. An edge subnet is used for efficient texture processing and detail preservation. The subnet is a full-surface network based on VGG-16 and can be easily integrated into any architecture. Based on the KITTI dataset, it is demonstrated that the fine details are recognized more accurately.

Brandao et al. [65] focused on the feature extraction component and demonstrated that image convolution improves the features used to find point matches. After feature extraction using the Siamese architecture, the features are combined according to their relative bias. The correlation between the features for each mismatch is computed using a simple two-layer correlation architecture. The proposed architecture achieved a 2.5-times lower error rate than the MC-CNN-acrt architecture.

Figure 19 illustrates the development chronology of stereo-matching methods using Siamese architecture. Schematic representation of stereo-matching methods is presented in Figure 20.



**Figure 19.** Timeline and links for major works in SNN stereo matching methods. The following works are mentioned: Zbontar and LeCun (2015a) [56]; Zbontar and LeCun (2015b) [57]; Zagoruyko and Komodakis (2015) [58]; Luo et al. (2016) [60]; Chen et al. (2016) [61]; Khamis et al. (2018) [62]; Kendall et al. (2017) [63]; Song et al. (2018) [64]; Brandao et al. (2019) [65]. The summaries of these works are provided in Section 4.7.

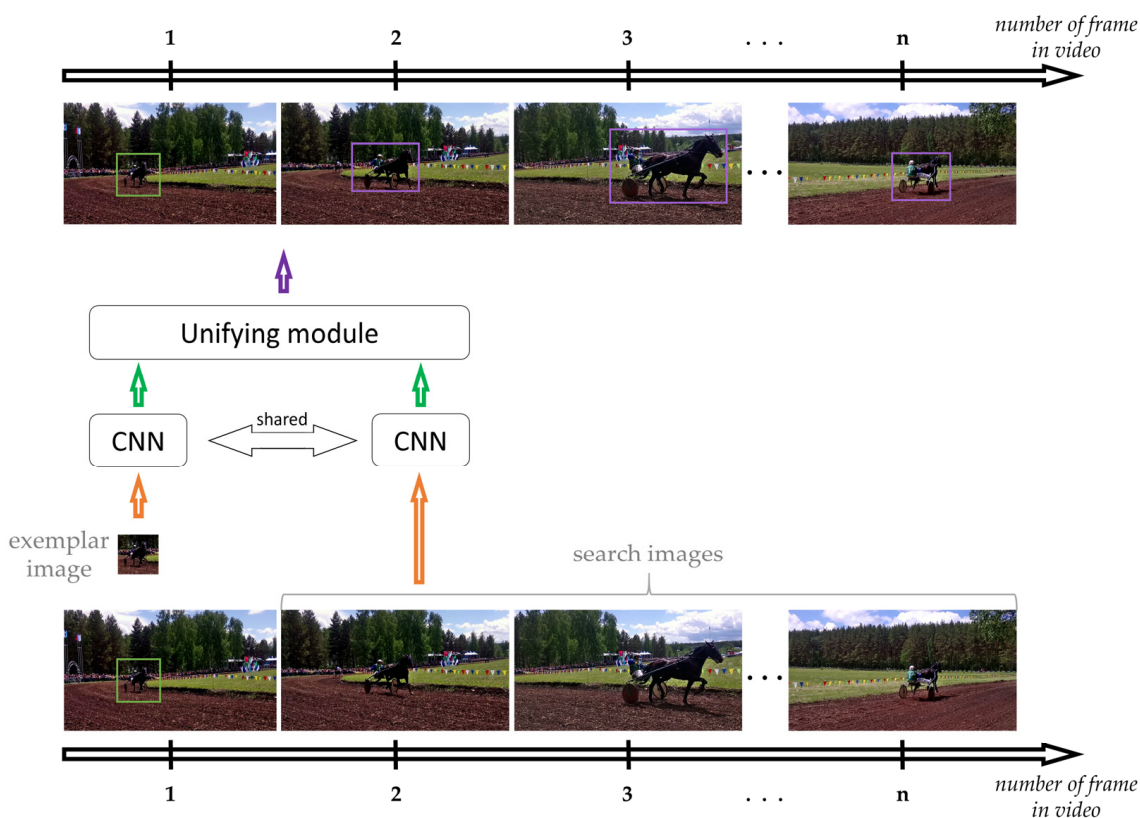
Method	Input data preprocessing	Input data sizes	Simplified diagram or architecture name of SNN branch	Similarity measuring / Unifying module	Loss function
Zbontar and LeCun (2015a)	grayscale, normalize	9×9 patches			Binary cross-entropy loss
Zbontar and LeCun (2015b)		9×9 or 11×11 patches	 		
Zagoruyko and Komodakis (2015)	2-channel network	64×64 patches			Hinge loss
	branch network	64×64 and 32×32 patches			
Luo et al. (2016)	normalize	37×37 patches and varying image sizes			Cross-entropy loss
Chen et al. (2016)	grayscale	13×13 patches			Euclidian loss

**Figure 20.** Schematic representation of preprocessing algorithms, input data size, SNN branch architectures, algorithms for similarity measurement (or merging modules), and loss functions for described stereo matching methods. The designations of the architecture layers are presented in Table 2. The following works are mentioned: Zbontar and LeCun (2015a) [56]; Zbontar and LeCun (2015b) [57]; Zagoruyko and Komodakis (2015) [58]; Luo et al. (2016) [60]; Chen et al. (2016) [61]; Khamis et al. (2018) [62]; Kendall et al. (2017) [63]; Song et al. (2018) [64]; Brandao et al. (2019) [65].

Despite the widespread use of the stereo matching tasks and a large amount of research in this area, accurate stereo matching is still challenging. Because of the varying amount of detail in different scenes, the stereo matching problem is not suitable for rigidly deterministic static algorithms. Despite this, neural networks are good at solving problems with implicit parameters and are suitable for solving such challenges. Also, deep learning allowed researchers to make significant progress in solving stereo matching tasks. Thus, Siamese networks have achieved excellent results in stereo matching tasks with increasing computational power, the network architectures and computational functions complexity, and by applying auxiliary solutions to SNN architectures.

#### 4.8. Visual Object Tracking

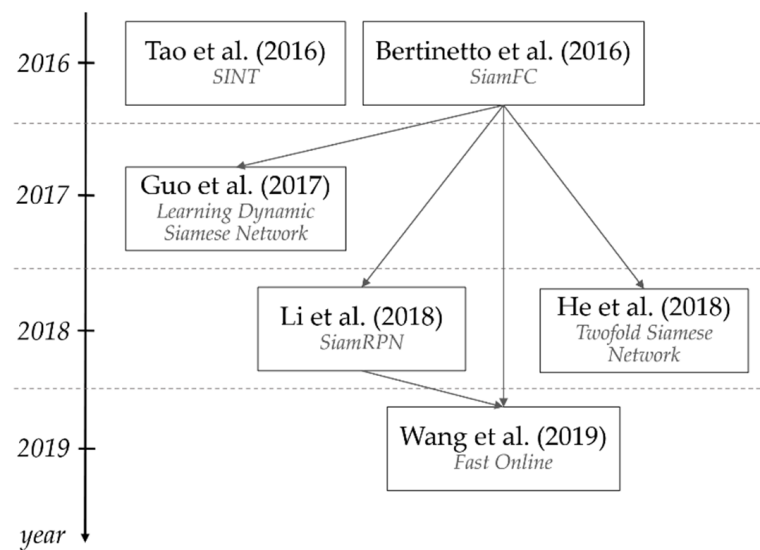
Visual object tracking (VOT) methods solve the task of estimating or predicting the moving object position in video. It is commonly used in automatic surveillance, vehicle navigation, video tagging, action recognition, etc. VOT systems first establish the object of interest’s location in the first video frame and extract an exemplar image from it. Then the system evaluates the exemplar position in all subsequent frames with the highest possible accuracy. A distinction is made between online and offline object tracking. In online tracking, video stream is analyzed in real time, i.e., only information from previous moments is used. In offline tracking, a video recording is analyzed, which allows use of the information about past frames and about subsequent moments. A simplified scheme of VOT systems operation is shown in Figure 21.



**Figure 21.** VOT systems operation. The reference image (exemplar image), containing the target object (the green color of the bounding box), is extracted from the first video frame. Subsequent frames (search images) come together with the reference image as input data for the Siamese architecture. Such architecture outputs the target object’s coordinates in the subsequent frames (violet bounding box).

Figure 22 presents a chronological diagram of VOT methods described in this paper (the most cited publications, which can be called the founders of VOT methods, are highlighted).

Tao et al. [66] developed the Siamese instance search tracker (SINT). Since tracking also has similarities with the object localization problem, the authors were inspired by [67] and included a ROI pooling layer in the proposed architecture for fast processing of multiple regions in a single frame. Also, the authors demonstrated the ability of the Siamese architecture and the region-of-interest matching function to track objects that were not used for training with no additional adaptations.



**Figure 22.** Timeline and links for major works in VOT techniques involving symmetrical neural networks. The following works are mentioned: Tao et al. (2016) [66]; Bertinetto et al. (2016) [68]; Li et al. (2018) [69]; Wang et al. (2019) [70]; Guo et al. (2017) [71]; He et al. (2018) [72]. The summaries of these works are provided in Section 4.8.

Bertinetto et al. [68] developed the SiamFC system for solving object tracking tasks, introducing a correlation layer as part of the merging module. The proposed architecture is trained to find the reference image in a larger image. It is achieved by calculating the mutual correlation of two feature maps obtained from the outputs of each Siamese structure branch. Then, the score map is obtained. The score map reveals information about the tracking object's location relative to the center of the image but does not solve the problem of its localization.

Li et al. [69] continued the idea of [68] and proposed the Siamese region proposal network (SiamRPN). This architecture contains a region proposal subnetwork (RPS) as a unifying module. RPS comprises two independent branches: binary classification and bounding box coordinate regression. In each branch, a pair-wise correlation is performed. The bounding box coordinates around the tracked object are the output of the proposed architecture.

Class-agnostic binary segmentation masks of the target object are proposed in the SiamMask system [70]. The authors propose a SiamRPN tracker extension for [69] by adding a branch for a pixel-wise binary mask. Also, a loss function for training of this branch is introduced. Authors of SiamMask claim that, after training, this system can work online, creating class-independent object segmentation masks and rotating bounding boxes.

Guo et al. [71] proposed a dynamic Siamese network (DSiam) for real-time object tracking. This system can adapt quickly to temporal variations in foreground and background by introducing target appearance variation transformation in the exemplar image branch and background suppression transformation in the search image branch. The authors also propose to use element-wise multi-layer fusion as a unifying module.

In [72] a dual SA-Siam network for real-time object tracking is proposed. Each Siamese architecture subnet here comprises a semantic branch (S-Net) and an appearance branch (A-Net). These branches are trained separately, so that appearance features for similarity search and semantic features for classification task complement each other. To improve the semantic branch's discriminatory ability, the authors proposed supplementing it with a channel attention module. This module comprises a max pooling layer, a multilayer perceptron, and a sigmoidal function.

Figure 23 illustrates a simplified structure of the features of the considered VOT techniques. In the figure below, the image preprocessing comprises only cropping and





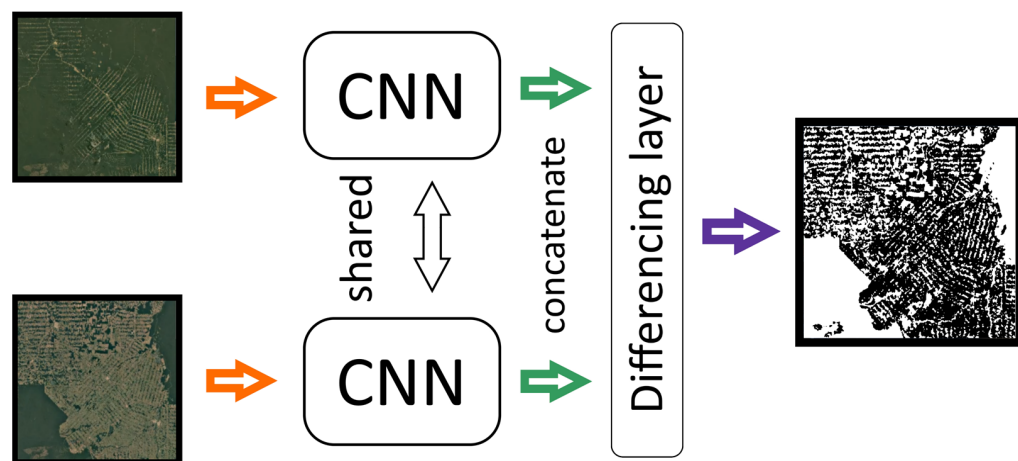


Figure 24. SNN application for change detection in images.

The main difference between the original SNN use for change detection is not limited to change or similarity detection but extended to the localization of detected changes. In [74] change detection accuracy exceeds 89%. The chronology of the publications along with the links between the main works is shown in Figure 25.

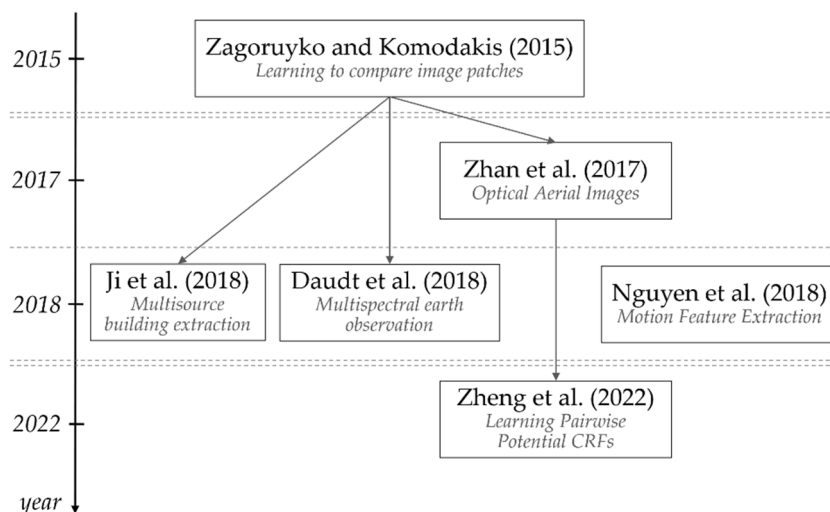


Figure 25. Timeline and links for major works in change detection techniques based on SNN. The following works are mentioned: Zagoruyko and Komodakis (2015) [58]; Zhan et al. (2017) [73]; Daudt et al. (2018) [74]; Ji et al. (2018) [75]; Zheng et al. (2022) [76]; Nguyen et al. (2018) [77]. The summaries of these works are provided in Section 4.9.

Figure 25 illustrates that the development paused for 12 years, then continued in 2017 with a new popularity wave beginning. One can see that the work of [58] was fundamental in this direction, later works [73–75] are based on the positions of [58], and the work of [76] is their consequence. Thus, we can conclude that there is no research atomization in this field, and the original approach proposed in [58] was proved useful. In [75], data augmentation based on the image comparisons obtained in different regions of the electromagnetic radiation spectrum is used for the analysis. Such augmentation has been shown to improve the accuracy and completeness of change detection.

Triplet networks for change detection on images are proposed to be used only in the known work [77]. High change-detection reliability on images is shown (average F-measure exceeds 0.84), but the comparison was made only with techniques based on different mathematical approaches. Nguyen et al. [77] did not make a comparison of their

proposal with traditional Siamese neural networks. The pre-processing consists of motion feature detection applying the neural network (MF Network). In the triplet part, the authors use a CNN with alternating convolutional and ReLU layers, and a hinge loss function.

## 5. Materials and Methods

Optimal (in terms of speed and power efficiency) implementation of large symmetrical neural networks at the hardware level deserves a separate consideration. For a long time, when solving urgent tasks in computer vision and voice recognition, a transition to specialized classes of microprocessors and coprocessors (often manufactured as a specialized integrated circuit) has been occurring. This is because the software implementation of artificial neural networks has several inherent drawbacks (for example, the resource-intensive computation of non-linear neuron activation functions and their derivatives) [78].

A well-known example of a specialized neuromorphic processor is IBM's TrueNorth processor, developed back in 2014 [79]. It has one million neurons and 256 million synaptic connections. A more recent example is the systolic array comprising arithmetic-logic devices (ALUs), which is the basis of Google Tensor processors [80]. A first-generation model, TPU1 has a  $256 \times 256$  ALU with a clock frequency of 700 MHz; the processor's performance is 92 TOPS. Power consumption of this processor is  $P \sim 37$  W, which corresponds to an energy efficiency  $E \sim 400$  fJ/operation. Second and third generations of Tensor processors (TPU2, TPU3) have reduced the systolic array size to  $128 \times 128$ , and the processing bit size to 16 bits. Each chip has two cores with one (TPU2) or two (TPU3) systolic arrays. The Tensor processor element (unit) board combines four chips. Performance per chip is 45 TFLOPS and 105 TFLOPS for the 2nd and 3rd generation chips, respectively. One Tensor element performance for the 2nd and 3rd generation is 180 TFLOPS and 420 TFLOPS, respectively. For resource-intensive neural network computations, the Tensor elements are combined into a single high-performance parallel system. However, hardware solutions based on digital signal processors and video chips also face problems when calculating nonlinear activation functions and their derivatives [81].

One of the promising areas of neuromorphic processor development is the use of memristors (resistors with memory). In the hybrid architecture, the memristor matrix plays the role of the synaptic connections, replacing the digital memory, and is above the CMOS logic layers in which the neurons are implemented. The drawbacks for these implementations are relatively high power consumption ( $P_{op} \sim 10^{-4}$  W, which corresponds to the total power consumption of the memristor matrix,  $P \sim 25.6$  kW for 256M synapses) and poor repeatability of the individual memristors, which can lead to low calculation accuracy [82,83].

Another direction of analog neuromorphic processor implementation is macroscopic quantum technologies. For example, neural networks using superconductivity and Josephson effects are characterized by fast performance ( $10^{10}$  vs.  $10^3$  impulses/neuron/s in superconducting NN in comparison with biological NN). Also, energy efficiency (power consumption at  $\sim 0.1$  mW) is an advantage [84–86].

It seems possible to estimate the performance of superconducting Tensor processors (STPUs) built on a standard element library. Considering its scaling within the intended improvement of Josephson junction production technology, STPUs built according to TPU3 architecture will have two orders of magnitude higher performance ( $10^{16}$  operations per second on one chip) and three orders of magnitude better energy efficiency ( $\sim 500$  aJ per operation) compared with the last generation TPU processor. Taking the cooling penalty into account, the energy efficiency advantage would be one order of magnitude. Combining four processor chips on a single board and then combining 25 boards in a rack, similar to Google's approach, one can provide superconducting computing performance of  $10^{18}$  operations per second.

## 6. Conclusions

The conducted survey has shown that symmetry plays an important role in many neural networking applications these days. Two main architectures of symmetrical neural networks are known to date: Siamese and triplet. These architectures may be used with various completing neural modules depending on the area of application and on the task being solved.

The survey is structured according to the most important and promising areas of application of symmetrical neural networks. We have identified nine areas from the “sea of publication activity” and examined them in detail. They are signature verification, speech verification, semantic text analysis, image retrieval, face analysis, re-identification, visual object tracking, stereo matching, and change detection. The selection criteria are the frequency of appearance of new important results and the citations of research, supported by the expert assessment of the authors of this study. The presence of impressive practical results and the continuation of research in the development of architectures and training methods for symmetrical neural networks is the common feature of most of the highlighted areas.

In the fields of signature verification and semantic text analysis, there are groups of interrelated studies, as well as a number of isolated works. At the same time, the quality of the results in all works is high. It is shown that existing SNNs demonstrate superiority over most competitors in signature verification tasks. For semantic text analysis methods using a Siamese structure, we can observe a rapid development from single-layer models with LSTM to complex, diverse multilevel models. The recurrent models’ application, in combination with many modern neural network solutions, breaks new ground in semantic text analysis.

In speech verification many isolated works are noted since 2011. The proposed methods provide high accuracy, but Siamese neural networks in the tasks of speech verification have not yet found wide popularity among the authors.

In the field of facial analysis, all works have a common ancestor from 2005, and do not refer to each other at all due to the development of different aspects laid down in the work of the ancestor of the direction. There is a trend of increasing complexity in SNN architectures with increasing input data and reducing the number of image preprocessing operations, indicating a gradual transition to automatic face verification methods. However, there are few methods for face comparison using the similarity learning concept.

In the areas of image retrieval and re-identification there is a very significant degree of isolation of studies, which indicates the initial, exploratory stage of research. At the same time, we can conclude that the image retrieval and re-identification tasks are paid much attention and new symmetrical neural networking techniques and architectures are being developed.

In stereo matching, major research is complete by 2019, with the most important problems of network architecture and learning metrics solved. Siamese networks have achieved excellent results in stereo matching tasks with increasing computational power, the network architecture and computational function complexity, and by applying auxiliary solutions to SNN architectures.

In the field of visual object tracking [87], isolated studies are almost absent, all works are interconnected and fit into the general paradigm. The most active research in this area began to fade after 2019. Analysis of publications describing VOT methods shows that this field is rapidly developing at present. However, the most popular target object tracking techniques are based on feature map correlations that are formed by fully convolutional neural networks jointed into the Siamese structure.

In the area of change detection, the most promising is the cluster of interconnected scientific research, which emerged in 2015 and is still developing. Triplet networks are almost out of consideration in this research area, the authors mostly rely on Siamese architectures just varying the loss functions and output layers.

Concerning the possible hardware implementation, we conclude that Tensor processors may be the most promising hardware platform for implementing symmetrical neural networks (including those based on superconducting technologies for especially hard computational tasks).

**Author Contributions:** Individual contributions to this paper: conceptualization, O.I. and M.T.; methodology, M.T., O.I. and V.Z.; search and analysis, O.I., V.Z. and M.T.; writing—original draft preparation, O.I. and V.Z.; writing—review and editing, M.T. and N.K.; supervision, M.T. and N.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This concept was developed with the support of the Russian Science Foundation Grant (RSF) No. 18-72-10118. The work of N.K. was supported by the RFBR grant 20-07-00952.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data is contained within the article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Chicco, D. Siamese neural networks: An overview. In *Artificial Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 73–94.
2. Ondrašovič, M.; Tarábek, P. Siamese Visual Object Tracking: A Survey. *IEEE Access* **2021**, *9*, 110149–110172. [[CrossRef](#)]
3. Nandy, A.; Haldar, S.; Banerjee, S.; Mitra, S. A survey on applications of Siamese neural networks in computer vision. In *Proceedings of the 2020 International Conference for Emerging Technology (INCET)*, Belgaum, India, 5–7 June 2020; pp. 1–5.
4. Kaya, M.; Bilge, H.Ş. Deep metric learning: A survey. *Symmetry* **2019**, *11*, 1066. [[CrossRef](#)]
5. Baldi, P.; Chauvin, Y. Neural Networks for Fingerprint Recognition. *Neural Comput.* **1993**, *5*, 402–418. [[CrossRef](#)]
6. Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; Shah, R. Signature verification using a “Siamese” time delay neural network. *Adv. Neural. Inf. Process Syst.* **1993**, *6*, 737–744. [[CrossRef](#)]
7. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 539–546. [[CrossRef](#)]
8. Hoffer, E.; Ailon, N. Deep Metric Learning Using Triplet Network. In *International Workshop on Similarity-Based Pattern Recognition*; *Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2015; pp. 84–92. [[CrossRef](#)]
9. Ahrabian, K.; BabaAli, B. Usage of autoencoders and Siamese networks for online handwritten signature verification. *Neural Comput. Appl.* **2018**, *31*, 9321–9334. [[CrossRef](#)]
10. Tolosana, R.; Vera-Rodriguez, R.; Fierrez, J.; Ortega-Garcia, J. Exploring Recurrent Neural Networks for On-Line Handwritten Signature Biometrics. *IEEE Access* **2018**, *6*, 5128–5138. [[CrossRef](#)]
11. Dey, S.; Dutta, A.; Toledo, J.I.; Ghosh, S.K.; Lladós, J.; Pal, U. Signet: Convolutional Siamese network for writer independent offline signature verification. *arXiv* **2017**, arXiv:1707.02131.
12. Jagtap, A.B.; Sawat, D.D.; Hegadi, R.S.; Hegadi, R.S. Verification of genuine and forged offline signatures using Siamese Neural Network (SNN). *Multimed. Tools Appl.* **2020**, *79*, 35109–35123. [[CrossRef](#)]
13. Ruiz, V.; Linares, I.; Sanchez, A.; Velez, J.F. Off-line handwritten signature verification using compositional synthetic generation of signatures and Siamese Neural Networks. *Neurocomputing* **2019**, *374*, 30–41. [[CrossRef](#)]
14. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *NIPS* **2012**, *25*, 1097–1105. [[CrossRef](#)]
15. Hameed, M.M.; Ahmad, R.; Kiah, M.L.M.; Murtaza, G. Machine learning-based offline signature verification systems: A systematic review. *Signal Process. Image Commun.* **2021**, *93*, 116139. [[CrossRef](#)]
16. Chen, K.; Salman, A. Learning speaker-specific characteristics with a deep neural architecture. *IEEE Trans. Neural Netw.* **2011**, *22*, 1744–1756. [[CrossRef](#)]
17. Chen, K.; Salman, A. Extracting Speaker-Specific Information with a Regularized Siamese Deep Network. *NIPS* **2011**, *2011*, 298–306.
18. Lu, X.; Shen, P.; Tsao, Y.; Kawai, H. Siamese Neural Network with Joint Bayesian Model Structure for Speaker Verification. In *Proceedings of the 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Tokyo, Japan, 14–17 December 2021.
19. Soleymani, S.; Dabouei, A.; Iranmanesh, S.M.; Kazemi, H.; Dawson, J.; Nasrabadi, N.M. Prosodic-enhanced siamese convolutional neural networks for cross-device text-independent speaker verification. In *Proceedings of the 2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, Redondo Beach, CA, USA, 22–25 October 2018; pp. 1–7.
20. Salehghaffari, H. Speaker verification using convolutional neural networks. *arXiv* **2018**, arXiv:1803.05427.

21. Zhang, Y.; Yu, M.; Li, N.; Yu, C.; Cui, J.; Yu, D. Seq2seq attentional siamese neural networks for text-dependent speaker verification. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 6131–6135.
22. Wang, J.; Qin, Y.; Peng, Z.; Lee, T. Child Speech Disorder Detection with Siamese Recurrent Network Using Speech Attribute Features. In Proceedings of the INTERSPEECH 2019, Graz, Austria, 15–19 September 2019; pp. 3885–3889.
23. Wang, M.; Deng, W. Deep Face Recognition: A Survey. *Neurocomputing* **2020**, *429*, 215–244. [\[CrossRef\]](#)
24. Khalil-Hani, M.; Sung, L.S. A convolutional neural network approach for face verification. In Proceedings of the 2014 International Conference on High Performance Computing & Simulation (HPCS), Bologna, Italy, 21–25 July 2014. [\[CrossRef\]](#)
25. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015. [\[CrossRef\]](#)
26. Li, L.; Feng, X.; Wu, X.; Xia, Z.; Hadid, A. Kinship Verification from Faces via Similarity Metric Based Convolutional Neural Network. In *Image Analysis and Recognition*; Springer: Cham, Switzerland, 2016; pp. 539–548. [\[CrossRef\]](#)
27. Bianco, S. Large Age-Gap face verification by feature injection in deep networks. *Pattern Recognit. Lett.* **2017**, *90*, 36–42. [\[CrossRef\]](#)
28. Borghi, G.; Pini, S.; Vezzani, R.; Cucchiara, R. Driver Face Verification with Depth Maps. *Sensors* **2019**, *19*, 3361. [\[CrossRef\]](#)
29. Xu, X.; Zhang, L.; Duan, C.; Lu, Y. Research on Inception Module Incorporated Siamese Convolutional Neural Networks to Realize Face Recognition. *IEEE Access* **2020**, *8*, 12168–12178. [\[CrossRef\]](#)
30. Heidari, M.; Fouladi-Ghaleh, K. Using Siamese Networks with Transfer Learning for Face Recognition on Small-Samples Datasets. In Proceedings of the 2020 International Conference on Machine Vision and Image Processing (MVIP), Tehran, Iran, 18–20 February 2020; pp. 1–4. [\[CrossRef\]](#)
31. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
32. Mueller, J.; Thyagarajan, A. Siamese recurrent architectures for learning sentence similarity. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 2786–2792.
33. Neculoiu, P.; Versteegh, M.; Rotaru, M. Learning Text Similarity with Siamese Recurrent Networks. In Proceedings of the 1st Workshop on Representation Learning for NLP, Berlin, Germany, 11 August 2016; pp. 148–157.
34. Shih, C.-H.; Yan, B.-C.; Liu, S.-H.; Chen, B. Investigating Siamese LSTM networks for text categorization. In Proceedings of the 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Kuala Lumpur, Malaysia, 12–15 December 2017. [\[CrossRef\]](#)
35. Pontes, E.L.; Huet, S.; Linhares, A.C.; Torres-Moreno, J.-M. Predicting the Semantic Textual Similarity with Siamese CNN and LSTM. *arXiv* **2018**, arXiv:1810.10641. [\[CrossRef\]](#)
36. Zhu, W.; Yao, T.; Ni, J.; Wei, B.; Lu, Z. Dependency-based Siamese long short-term memory network for learning sentence representations. *PLoS ONE* **2018**, *13*, e0193919. [\[CrossRef\]](#) [\[PubMed\]](#)
37. Chen, D.; Manning, C. A Fast and Accurate Dependency Parser using Neural Networks. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Stroudsburg, PA, USA, 2014; pp. 740–750.
38. Kumar, V.; Khattar, D.; Gairola, S.; Kumar Lal, Y.; Varma, V. Identifying Clickbait. In Proceedings of the 41st International ACM SIGIR Conference, Ann Arbor, MI, USA, 8–12 July 2018. [\[CrossRef\]](#)
39. Jiang, J.-Y.; Zhang, M.; Li, C.; Bendersky, M.; Golbandi, N.; Najork, M. Semantic Text Matching for Long-Form Documents. In Proceedings of the WWW Conference, San Francisco, CA, USA, 13 May 2019. [\[CrossRef\]](#)
40. Yin, G.; Liu, B.; Sheng, L.; Yu, N.; Wang, X.; Shao, J. Semantics Disentangling for Text-to-Image Generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019. [\[CrossRef\]](#)
41. Wang, F.; Kang, L.; Yi, L. Sketch-based 3D shape retrieval using Convolutional Neural Networks. In Proceedings of the IEEE Conference CVPR, Boston, MA, USA, 7–12 June 2015. [\[CrossRef\]](#)
42. Qi, Y.; Song, Y.-Z.; Zhang, H.; Liu, J. Sketch-based image retrieval via Siamese convolutional neural network. In Proceedings of the IEEE ICIP, Phoenix, AZ, USA, 25–28 September 2016. [\[CrossRef\]](#)
43. Bui, T.; Ribeiro, L.; Ponti, M.; Collomosse, J. Compact descriptors for sketch-based image retrieval using a triplet loss convolutional neural network. *Comput. Vis. Image Underst.* **2017**, *164*, 27–37. [\[CrossRef\]](#)
44. Lin, H.; Fu, Y.; Lu, P.; Gong, S.; Xue, X.; Jiang, Y.-G. TC-Net for iSBIR. In Proceedings of the 27th ACM International Conference MM, Nice, France, 21–25 October 2019. [\[CrossRef\]](#)
45. Cai, Y.; Li, Y.; Qiu, C.; Ma, J.; Gao, X. Medical Image Retrieval Based on Convolutional Neural Network and Supervised Hashing. *IEEE Access* **2019**, *7*, 51877–51885. [\[CrossRef\]](#)
46. Deepak, S.; Ameer, P.M. Retrieval of brain MRI with tumor using contrastive loss based similarity on GoogLeNet encodings. *Comput. Biol. Med.* **2020**, *125*, 103993. [\[CrossRef\]](#) [\[PubMed\]](#)
47. Zhang, K.; Qi, S.; Cai, J.; Zhao, D.; Yu, T.; Yue, Y.; Yao, Y.; Qian, W. Content-based image retrieval with a Convolutional Siamese Neural Network: Distinguishing lung cancer and tuberculosis in CT images. *Comput. Biol. Med.* **2022**, *140*, 105096. [\[CrossRef\]](#)
48. Yi, D.; Lei, Z.; Liao, S.; Li, S.Z. Deep Metric Learning for Person Re-identification. In Proceedings of the 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 34–39. [\[CrossRef\]](#)
49. McLaughlin, N.; Del Rincon, J.M.; Miller, P. Recurrent convolutional network for video-based person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1325–1334.

50. Varior, R.R.; Shuai, B.; Lu, J.; Xu, D.; Wang, G. A siamese long short-term memory architecture for human re-identification. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 135–153.
51. Zheng, M.; Karanam, S.; Wu, Z.; Radke, R.J. Re-identification with consistent attentive siamese networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5735–5744.
52. Mazzeo, P.L.; Libetta, C.; Spagnolo, P.; Distanto, C. A Siamese Neural Network for Non-Invasive Baggage Re-Identification. *J. Imaging* **2020**, *6*, 126. [[CrossRef](#)]
53. Schneider, S.; Taylor, G.W.; Kremer, S.C. Similarity learning networks for animal individual re-identification-beyond the capabilities of a human observer. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, Waikoloa, HI, USA, 4–8 January 2022; pp. 44–52.
54. Shen, Y.; Xiao, T.; Li, H.; Yi, S.; Wang, X. Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1900–1909.
55. Scharstein, D.; Szeliski, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vis.* **2002**, *47*, 7–42. [[CrossRef](#)]
56. Zbontar, J.; LeCun, Y. Computing the stereo matching cost with a convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1592–1599.
57. Zbontar, J.; LeCun, Y. Stereo matching by training a convolutional neural network to compare image patches. *J. Mach. Learn. Res.* **2015**, *17*, 2287–2318.
58. Zagoruyko, S.; Komodakis, N. Learning to compare image patches via convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4353–4361.
59. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [[CrossRef](#)]
60. Luo, W.; Schwing, A.G.; Urtasun, R. Efficient deep learning for stereo matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5695–5703.
61. Chen, Z.; Sun, X.; Wang, L.; Yu, Y.; Huang, C. A deep visual correspondence embedding model for stereo matching costs. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 972–980.
62. Khamis, S.; Fanello, S.; Rhemann, C.; Kowdle, A.; Valentin, J.; Izadi, S. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 573–590.
63. Kendall, A.; Martirosyan, H.; Dasgupta, S.; Henry, P.; Kennedy, R.; Bachrach, A.; Bry, A. End-to-end learning of geometry and context for deep stereo regression. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 66–75.
64. Song, X.; Zhao, X.; Hu, H.; Fang, L. Edgestereo: A context integrated residual pyramid network for stereo matching. In Proceedings of the Asian Conference on Computer Vision, Perth, Australia, 2–6 December 2018; pp. 20–35.
65. Brandao, P.; Mazomenos, E.; Stoyanov, D. Widening siamese architectures for stereo matching. *Pattern Recognit. Lett.* **2019**, *120*, 75–81. [[CrossRef](#)]
66. Tao, R.; Gavves, E.; Smeulders, A.W.M. Siamese Instance Search for Tracking. In Proceedings of the IEEE CVPR, Las Vegas, NV, USA, 27–30 June 2016. [[CrossRef](#)]
67. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE TPAMI* **2017**, *39*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
68. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-Convolutional Siamese Networks for Object Tracking. In *Computer Vision—ECCV 2016 Workshops*; Springer: Cham, Switzerland, 2016; pp. 850–865. [[CrossRef](#)]
69. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High Performance Visual Tracking with Siamese Region Proposal Network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [[CrossRef](#)]
70. Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H.S. Fast Online Object Tracking and Segmentation: A Unifying Approach. In Proceedings of the IEEE/CVF Conference CVPR, Long Beach, CA, USA, 15–20 June 2019. [[CrossRef](#)]
71. Guo, Q.; Feng, W.; Zhou, C.; Huang, R.; Wan, L.; Wang, S. Learning Dynamic Siamese Network for Visual Object Tracking. In Proceedings of the IEEE ICCV, Venice, Italy, 22–29 October 2017. [[CrossRef](#)]
72. He, A.; Luo, C.; Tian, X.; Zeng, W. A Twofold Siamese Network for Real-Time Object Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [[CrossRef](#)]
73. Zhan, Y.; Fu, K.; Yan, M.; Sun, X.; Wang, H.; Qiu, X. Change Detection Based on Deep Siamese Convolutional Network for Optical Aerial Images. *IEEE Geosci. Remote Sens. Lett.* **2015**, *14*, 1845–1849. [[CrossRef](#)]
74. Daudt, R.C.; Le Saux, B.; Boulch, A.; Gousseau, Y. Urban change detection for multispectral earth observation using convolutional neural networks. In Proceedings of the IGARSS 2018–2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 2115–2118.
75. Ji, S.; Wei, S.; Lu, M. Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 574–586. [[CrossRef](#)]
76. Zheng, D.; Wei, Z.; Wu, Z.; Liu, J. Learning Pairwise Potential CRFs in Deep Siamese Network for Change Detection. *Remote Sens.* **2022**, *14*, 841. [[CrossRef](#)]

77. Nguyen, T.P.; Pham, C.C.; Ha, S.V.U.; Jeon, J.W. Change Detection by Training a Triplet Network for Motion Feature Extraction. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 433–446. [[CrossRef](#)]
78. Meher, P.K.; Valls, J.; Juang, T.-B.; Sridharan, K.; Maharatna, K. 50 years of CORDIC: Algorithms, architectures, and applications. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2009**, *56*, 1893–1907. [[CrossRef](#)]
79. Merolla, P.A.; Arthur, J.V.; Alvarez-Icaza, R.; Cassidy, A.S.; Sawada, J.; Akopyan, F.; Jackson, B.L.; Imam, N.; Guo, C.; Nakamura, Y.; et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **2014**, *345*, 668–673. [[CrossRef](#)]
80. Kumar, S.; Wang, Y.; Young, C.; Bradbury, J.; Kumar, N.; Chen, D.; Swing, A. Exploring the limits of Concurrency in ML Training on Google TPUs. *Proc. Mach. Learn. Syst.* **2021**, *3*, 81–92.
81. Nickolls, J.; Dally, W.J. The GPU computing era. *IEEE Micro* **2010**, *30*, 56–69. [[CrossRef](#)]
82. Jeong, H.; Shi, L. Memristor devices for neural networks. *J. Phys. D Appl. Phys.* **2018**, *52*, 023003. [[CrossRef](#)]
83. Yao, P.; Wu, H.; Gao, B.; Tang, J.; Zhang, Q.; Zhang, W.; Yang, J.J.; Qian, H. Fully hardware-implemented memristor convolutional neural network. *Nature* **2020**, *577*, 641–646. [[CrossRef](#)] [[PubMed](#)]
84. Schegolev, A.; Klenov, N.; Soloviev, I.; Tereshonok, M. Learning cell for superconducting neural networks. *Supercond. Sci. Technol.* **2020**, *34*, 015006. [[CrossRef](#)]
85. Schneider, M.; Toomey, E.; Rowlands, G.; Shainline, J.; Tschirhart, P.; Segall, K. SuperMind: A survey of the potential of superconducting electronics for neuromorphic computing. *Supercond. Sci. Technol.* **2022**, *35*, 053001. [[CrossRef](#)]
86. Skryabina, O.V.; Schegolev, A.E.; Klenov, N.V.; Bakurskiy, S.V.; Shishkin, A.G.; Sotnichuk, S.V.; Napolskii, K.S.; Nazhestkin, I.A.; Soloviev, I.I.; Kupriyanov, M.Y.; et al. Superconducting Bio-Inspired Au-Nanowire-Based Neurons. *Nanomaterials* **2022**, *12*, 1671. [[CrossRef](#)]
87. Li, M.J.R.; Liu, Q.; Shi, Y.; Tlelo-Cuautle, E. High speed long-term visual object tracking algorithm for real robot systems. *Neurocomputing* **2021**, *434*, 268–284. [[CrossRef](#)]