Article

Convolutional Extreme Learning Machines: A Systematic Review

Iago Richard Rodrigues [1], Sebastião Rogério [2], Judith Kelner [1], Djamel Sadok [1] and Patricia Takako Endo [2]

[1] Centro de Informática, Universidade Federal de Pernambuco (UFPE); irrs@cin.ufpe.br, jk@cin.ufpe.br, jamel@cin.ufpe.br
[2] Universidade de Pernambuco (UPE); srsn@ecomp.poli.br, patricia.endo@upe.br
* Correspondence: irrs@cin.ufpe.br (I. R. Rodrigues); patricia.endo@upe.br (P. T. Endo).

Abstract: Many works have recently identified the need to combine deep learning with extreme learning to strike a performance balance with accuracy especially in the domain of multimedia applications. Considering this new paradigm, namely convolutional extreme learning machine (CELM), we present a systematic review that investigates alternative deep learning architectures that use extreme learning machine (ELM) for a faster training to solve problems based on image analysis. We detail each of the architectures found in the literature, application scenarios, benchmark datasets, main results, advantages, and present the open challenges for CELM. We follow a well structured methodology and establish relevant research questions that guide our findings. We hope that the observation and classification of such works can leverage the CELM research area providing a good starting point to cope with some of the current problems in the image-based computer vision analysis.

Keywords: Convolutional extreme learning machine; Deep learning; Multimedia analysis

1. Introduction

Due to the growth of image analysis-based applications, researchers adopted deep learning to develop intelligent systems that provide learning tasks in computer vision, image processing, text recognition, and other signal processing problems. Deep learning architectures are generally a good solution for learning on large-scale data, surpassing classic models that were once state of the art in multimedia problems [1].

Unlike classic approaches to pattern recognition tasks, convolutional neural networks (CNNs), a type of deep learning, can perform the process of extracting features and at the same time recognize these features. CNNs can process data stored as multi-dimensional arrays (1D, 2D, and so on). They extract meaningful abstract representations from raw data [1], such as images, audio, text, video, and so on. CNN's also have received attention in the last decade due to their success obtained in fields such as image classification [2], object detection [3], semantic segmentation [4], and medical applications that support a diagnosis by signals or images [5].

Despite its benefits, CNNs also suffers from some challenges: it incurs a high computational cost, impacting directly on training and inference times. Classification time is an issue for real-time applications that tolerate a minimal loss of information. Another challenge is the long training and testing times if we consider a computer with limited hardware resources. Other problems are local minimum, intensive human intervention, and vanishing gradients [6]. It is, therefore, necessary to investigate alternative approaches that may extract deep feature representation and, at the same time, reduce the computational cost.

Extreme learning machine (ELM) is a type of single-layer feed-forward neural network (SLFN) [7] that provides a faster convergence training process and does not require a series of iterations to adjust the weights of the hidden layers. According to [8], "*seems that ELM performs better than other conventional learning algorithms in applications with higher noise*", presenting similar or better generalizations in regression and classification tasks. Unlike others, a ELM model executes a single hidden layer of neurons with random feature mapping, providing a faster learning execution. The low computational complexity attracted a great deal of attention from the research community, especially for high dimensional and large data applications [9].

Based on the strengths of CNNs and ELMs, a new neural network paradigm was proposed, the convolutional extreme learning machine (CELM) [10]. CELMs are quick-training CNNs that avoid gradient calculations for updating the network weights. Filters are efficiently defined for the feature extraction step, and least-squares obtain weights in the classification stage's output layer through an ELM network architecture. In most cases, the accuracy achieved by CELMs is not the best one [10]. However, the results are very competitive compared to those obtained by convolutional networks, not only in terms of accuracy but also in training and inference time.

Some works in the literature have presented a survey on ELM from different perspectives. Huang et al. [11] present a survey of ELM and its variants. They focused on describing the fundamental design principles and learning theories. The main ELM variants presented by the authors are (i) batch learning mode of ELM, (ii) fully complex ELM, (iii) online sequential ELM, (iv) incremental ELM, and (v) ensemble of ELM. Cao et al. [8] present a survey on ELM while mainly considering high dimensional and large-data applications. The works in the literature are classified into image processing, video processing, and medical signal processing. Huang et al. [12] present trends in ELM, including ensembles, semi-supervised learning, imbalanced data, and applications such as computer vision and image processing. Salaken et al. [13] explore ELM in conjunction with transfer learning algorithms. Zhang et al. [14] present current approaches based on multilayer ELM (ML-ELM) and its variants compared to classical deep learning.

However, despite some ELM surveys, no work focuses specifically on CELM studies from the authors' knowledge. Therefore, differently from the existing literature, we present a systematic review that concentrates on CELM applied in the context of analysis.

The works considered in this systematic review should attend two main concepts in their purposes: (i) usage of deep feature representation through convolution operations and (ii) usage of ELM aiming to achieve fast feature learning in/after the convolution stage. We discuss the proposed architectures, the application scenarios, the benchmark datasets, the principal results and advantages, and the open challenges in the CELM field.

The rest of this work is organized as follows: Section 2 presents the methodology adopted to conduct this systematic review. The overview of primary studies of this systematic review are presented at Section 3. The answers for each research question defined in the systematic review protocol are described in sections 4, 5, 6, and 7. Finally, we conclude this work in Section 8.

2. Methodology

To perform the systematic review, we adopted the methodology previously used by Endo et al. [15] and Coutinho et al. [16]. The mentioned systematic review protocol was originally inspired by the classic protocol proposed by Kitchenham [17]. Fig 1 illustrates the methodology adopted in this work. Next, we explain each of these steps.

2.1. Identify need of review

Due to the growth of image and big data applications, both academia and industry used deep learning to analyze data and extract relevant information. For large networks, deep learning architectures suffer drawbacks such as high computational cost, slow convergence, vanishing gradients, and hardware limitations for training.

In this systematic review, we mainly investigate the use of CELM as a viable alternative for deep learning architectures while guarantying quick-training and avoiding gradient

doi:10.20944/preprints202104.0753.v1

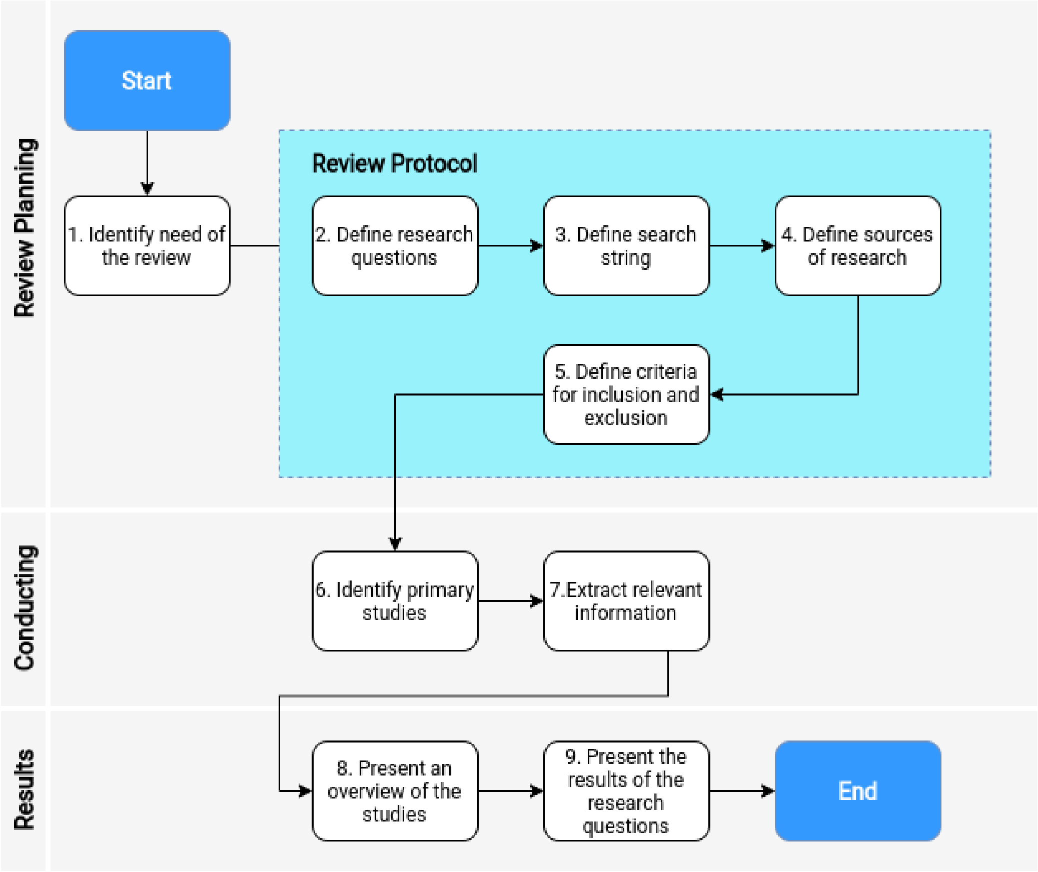

Review Planning
Start
Review Protocol
1. Identify need of the review
2. Define research questions
3. Define search string
4. Define sources of research
5. Define criteria for inclusion and exclusion
Conducting
6. Identify primary studies
7.Extract relevant information
Results
8. Present an overview of the studies
9. Present the results of the research questions
End

Figure 1. Methodology to select papers in this systematic review.

calculations necessary for updating the network's weights. In recent years, CELM's have solved some of the leading deep learning issues while maintaining a reasonable quality of the solutions in many applications.

2.2. *Define research questions*

We initiate with the definition of four research questions (RQ) related to our topic of study. Our objective is to answer these research questions to raise a discussion of the current state of the art in the usage of CELM in the specific domain of image analysis. The research questions are:

- RQ 1: What are the most common problems based on image analysis and datasets analyzed under CELM?
- RQ 2: How are defined the CELM architectures in the analyzed works?
- RQ 3: Which are the main findings when applying CELM in problems based on image analysis?
- RQ 4: What are the main open challenges in applying CELM to problems based on image analysis?

2.3. *Define search string*

To find articles related to our RQs, it was necessary to define a suitable search string to be used in the search sources adopted. To create such a search string, we defined terms and synonyms related to this research's scope. The search string defined was *"(("ELM" OR "extreme learning machine" OR "extreme learning machines") AND ("image recognition" OR "image classification" OR "object recognition" OR "object classification" OR "image segmentation"))"*.

2.4. Define sources of research

We adopted the following traditional search sources (databases) to get the articles: IEEE Xplore[1], Springer Link[2], ACM Digital Library[3], Science Direct[4], SCOPUS[5], and Web of Science[6].

Since we consider the four primary databases (IEEE Xplore, Springer Link, ACM DL, and Science Direct) and two meta-databases (SCOPUS and Web of Science), we first selected the articles from the primary databases because the meta-databases provided some duplicate results.

2.5. Define criteria for inclusion and exclusion

We defined criteria for the inclusion and exclusion of articles in this systematic review aiming to obtain only articles within the scope of this research. The criteria are:

- Primary studies published in peer-reviewed journals or conferences (congress, symposium, workshop, etc.);
- Works that answer one or more RQs defined in this systematic review;
- Works published from 2010 to 2020;
- Works published in English;
- Works accessible or freely available (using University proxy) from the search sources used in this project.

2.6. Identify primary studies

We identified the primary studies according to the inclusion and exclusion criteria defined in Section 2.5.

2.7. Extract relevant information

We extracted relevant information from the primary studies by reading the entire paper and answering the RQs.

2.8. Present an overview of the studies

In this step, we will present a general summary of the primary studies selected in the systematic review. The overview information includes a percentage of the year of publication of the articles and the database from which it was obtained. Section 3 presents the overview of the studies.

2.9. Present the results of the research questions

Considering the research questions defined in Section 2.2, we will present the answers found with the analysis of the selected articles. The answer to the defined research questions characterizes the main contribution of this systematic review. The results of this step are presented in sections 4, 5, 6, and 7.

3. Overview of the primary studies

Table 1 presents the number of works before and after applying the inclusion and exclusion criteria. A total of 2,220 articles were returned from the six databases. After removing duplicate articles and applying inclusion and exclusion criteria, 83 articles remained, which correspond to 3.74% of the total articles found in the search.

We selected 30 papers from Springer Link, and this is the database with more primary studies returned. IEEE Xplore returned the second-highest number of primary studies,

[1] https://www.ieeexplore.ieee.org/
[2] https://link.springer.com/
[3] https://dl.acm.org/
[4] https://www.sciencedirect.com/
[5] https://www.scopus.com/
[6] https://www.webofknowledge.com

Table 1: Search results obtained before and after refinement by inclusion and exclusion criteria.

Database	Original search	After primary studies identification
ACM DL	91	3
IEEE Xplore	123	19
Science Direct	54	6
Springer Link	992	30
SCOPUS	616	19
Web of Science	344	4

19. Science Direct and ACM returned 6 and 3 studies, respectively. Also, we can see the importance of using meta-databases in this study. The meta-databases also returned important works (19 in SCOPUS and 4 in Web of Science).

Regarding the primary studies identified, Fig. 2 illustrates the percentage of publication of these works per year. Although we establish a time range between 2010 to 2020, articles on CELM started to be published in 2015. A probable explanation for this is that there was the first consolidation of DL in the literature and multimedia applications in general. During this consolidation, several alternatives to conventional CNNs were proposed, such as CNN with many layers [18], residual CNN networks [19], networks with batch normalization [20], dropout [21], and other advances [22]. Besides, the literature sought alternatives for better generalization capacity and better training and classification time when CELM variations were proposed.

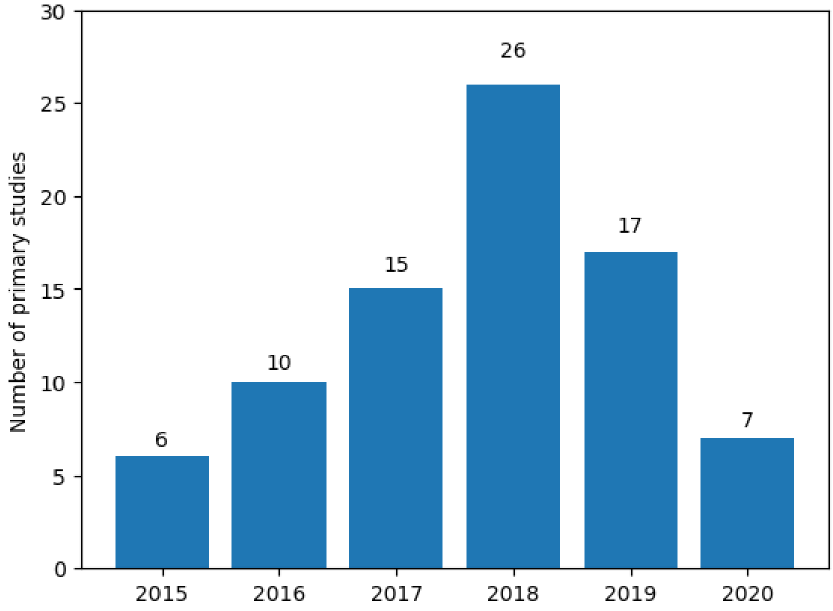


Figure 2. Articles distribution by publication year.

4. RQ 1: What are the most common problems based on image analysis and datasets analyzed under CELM?

From the primary studies, the main machine learning problems for multimedia analysis can be divided into two main groups: image classification and semantic segmentation.

Eighty works are related to image classification. Image classification is the process of labeling images according to information present in these images [2] and done by recognizing patterns. The classification process usually analyzes the image and associates it with a label describing an object. Image classification may be done through manual feature extraction + classical machine learning algorithms or deep learning architectures, which learn patterns in the feature extraction process.

Only one work [23] covers semantic segmentation. Semantic segmentation in images consists of categorizing each pixel present in the image [4]. The learning models are trained from ground truth information, which are annotations equivalent to each pixel's category pertinence of the input image. This model type's output is the segmented image, with each pixel adequately assigned to an object.

What may explain the high difference in the number of works for image classification instead of semantic segmentation is the triviality of implementing the CELM models for the first purpose. For the image classification task, the architectures are stacked with convolutional layers, pooling, and ELM concepts being placed sequentially (see more details in RQ 2). This fact facilitates the implementation of the CELM models.

Models for semantic segmentation need other concepts to be effective. In semantic segmentation, it is necessary to make predictions at the pixel level, which requires convolution and deconvolution steps to reconstruct the output images. These concepts may be the target by researchers in the future.

Note that object detection is also a common problem in the computer vision field, but we did not find works solving object detection using CELM concepts in this systematic review.

We found 19 different scenarios from the primary studies, but most of them contain four or fewer related works. This way, we highlight the six main application scenarios, and the others are demarcated in a single group (Others), as shown in Fig. 3. The six main CELM application scenarios found among the primary studies are: object recognition, remote sensing, medicine, handwritten digit recognition, RGB-D image recognition, and face recognition, totaling 69 articles, about 83% of the total primary studies.

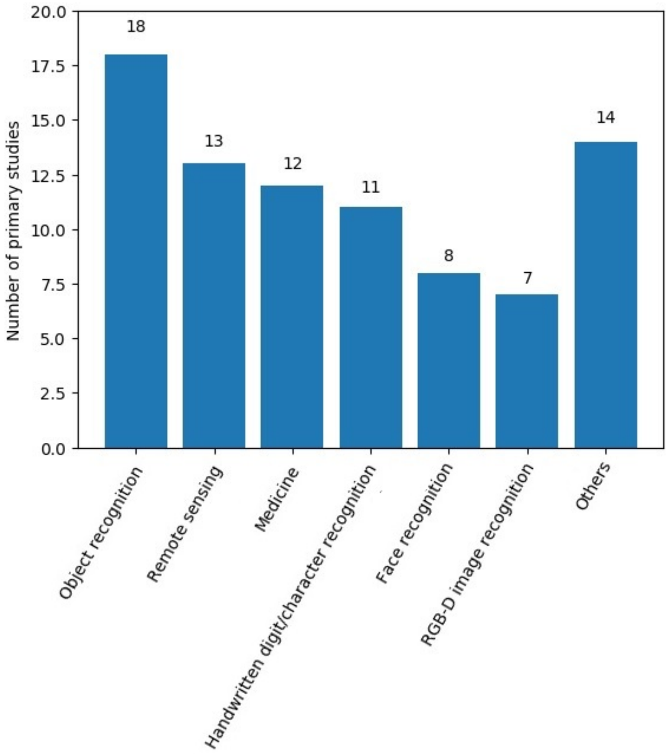


Figure 3. Main application scenarios of CELM.

4.1. *Object recognition*

Object recognition is one of the most common problems among the primary studies found in this systematic review. Object recognition consists of classifying objects in scenes and is not a trivial task. Generally, the dataset that makes up an object recognition problem comprises several elements divided by classes. The variation is realized in the object positions, lighting conditions, and so on. We have found 18 works dealing with object recognition, about 21% of the primary studies.

Among the primary studies, we found nine different object recognition datasets: NORB, CIFAR-10, CIFAR-100, Sun-397, COIL, ETH-80, Caltech, GERMS, and DR. These datasets, in general, have a wide range of classes, hampering the ability to generalize machine learning models. Therefore, if a proposed model obtains expressive results using large datasets for object recognition, there is strong evidence that this model presents a good generalization capacity.

Table 2 shows the reported datasets for object recognition and their respective references. Most works use NORB and CIFAR-10 datasets (representing more than 50% of usage). Note that some works use more than one dataset for training and testing their models. Next, we present a brief description of the main datasets found for object recognition.

Table 2: Datasets for object recognition reported in the primary studies.

Dataset	References
NORB	[24] [6] [25] [26] [27], [28] [29] [30]
CIFAR-10	[24] [27] [29] [31] [32], [33] [34] [35]
CALTECH	[36] [37] [35] [28]
COIL	[34] [29] [25]
CIFAR-100	[33] [35]
ETH-80	[34] [25]
SUN-397	[33]
GERMS	[38]
DR	[39]

The NYU Object Recognition Benchmark (NORB) dataset [40] is composed of 194,400 images that are pairs of stereo images from 5 generic categories under different angles, light setups, and pose. The variations consist of 36 azimuths, nine elevations, and six light setups.

Canadian Institute For Advanced Research (CIFAR-10) [41] is a dataset that contains 60,000 tiny images 32 × 32 size divided into 10 classes of objects, with 6,000 images per class. Also, CIFAR-100 contains 100 classes of objects, with 600 images per class. The default split protocol is 50,000 images for the train and 10,000 for the test.

The Columbia University Image Library (COIL) [42] is object image dataset. There are two main variations, such as: COIL-20, a dataset that contains 20 different classes of grayscale images, and COIL-100, which contains 100 classes of colored images. A total of 7,200 images compose the COIL-100 dataset.

Caltech-101 [43] is a dataset that contains 101 categories. There are 40 to 800 images per class, and most categories contain about 50 images with a size of 300 × 200. The last version of the dataset is Caltech-256, which contains 256 categories and 30,607 images.

ETH-80 [44] is a dataset composed of 8 different classes. Each class contains ten object instances, and 41 images compose each instance. There is a total of 3,280 images in the dataset.

4.2. Remote sensing classification

Remote sensing is information from a geospatial area acquired at a distance. The most common examples of remote sensing classification data are spectral images, which are different from ordinary RGB images since they carry data about infrared, ultraviolet, and so on. With this type of information, it is possible to obtain more detailed mapping of a remote sensing area. The other variation of data for remote sensing is called hyperspectral imaging [45], which in addition to spectrum information, also considers digital photographs. CELM has been used as an alternative solution in remote sensing classification because deep learning models generally require high processing for this type of application. In this systematic review, we reported a total of 13 works that applied CELM for remote sensing classification.

Table 3 shows seven datasets used for remote sensing classification found in our primary studies. The two main datasets are Pavia (8 works) and Indian pines (6 works), totaling about 60.9%. The other datasets are Salinas, MSTAR, UCM, AID, and R+N. Next, we present a brief description of the main datasets.

Table 3: Datasets for remote sensing classification reported in the primary studies.

Dataset	References
Pavia	[46] [47] [48] [49] [50] [51] [52] [53]
Indian Pines	[46] [47] [48] [54] [50] [51]
Salinas	[46] [47] [49] [53]
MSTAR	[55] [56]
UCM	[57]
AID	[57]
R+N	[57]

The Pavia dataset [58] is composed of nine different classes of scenes obtained by the ROSIS sensor[7] and the total number of spectral bands is 205. In the dataset, there are images of the size of 1096 × 1096 pixels and 610 × 610 pixels.

The Indian Pines dataset [58] consists of scenes collected by the AVIRIS sensor[8]. The data size corresponds to 145 × 145 pixels and 224 bands of spectral reflectance. The Indian Pines scenes contain scenes of agriculture and forests. There is also an immense amount of geographic data on houses, roads, and railways.

Like the Indian Pines dataset, the Salinas dataset [58] was collected by the 224-band AVIRIS sensor. Salinas dataset contains a high spatial resolution with 3.7-meter pixels. The area covered comprises 512 lines by 217 samples. The dataset contains 16 ground-truthed classes.

Moving and Stationary Target Acquisition and Recognition (MSTAR) is a dataset [59] that contains baseline X-band SAR imagery of 13 target types plus minor examples of articulation, obscuration, and camouflage. The Sandia National Laboratory collected the dataset and Defense Advanced Research [60].

4.3. *Medicine applications*

There is a growing increase in the number of machine learning applications for medicine. Most of them aim to identify patterns in imaging exams to support (not replace) the specialist. Generally, the data used is labeled by medical specialists in the study field of the disease to be identified. Applications of CELM models are made possible because they often surpass traditional models for the classification stage. All 12 works found in this systematic review aimed to provide support for decision-making in diagnosing various diseases.

Table 4 shows 12 applications of CELM for medicine reported in the primary studies returned, such as: tumor classification, anomalies detection, white blood cell detection, and so on. The application with most number of works is brain tumor classification [61], [62], [63]. Due to the variety of medical problems, the works do not use a common dataset, making it difficult to compare them.

4.4. *Handwritten digit and character recognition*

Similar to the object recognition problem, the handwritten digit and character recognition problem recurs in digital image processing, and pattern recognition benchmarking

[7] https://www.uv.es/leo/daisex/Sensors/ROSIS.htm
[8] https://aviris.jpl.nasa.gov/

Table 4: Applications in medicine and their datasets reported in the primary studies.

References	Approach	Dataset
[64]	Classification of digestive organs disease	Own dataset
[65]	Liver tumor classification	Elazig University Hospital
[66]	White blood cell detection	BCCD dataset
[67]	Histopathological image classification	ADL dataset
[68]	Cerebral microbleed diagnosis	Own dataset
[69]	Cervical cancer classification	Herlev dataset
[61]	Brain tumor classification	CGA-GBM database
[70]	Micro-nodules classification	LIDC/IDRI dataset
[62]	Brain tumor classification	Brain T1-weighed CE-MRI dataset
[63]	Brain tumor classification	Brain tumor MRI dataset
[71]	Classification of anomalies in the human retina	Duke and HUCM datasets
[72]	Hepatocellular carcinoma classification	ICPR 2014 HEp-2 cell dataset

[73]. Several works proposed digit and character recognition for applications such as handwriting recognition [73]. Handwritten digit or character recognition can be applied to several tasks: text categorization from images, classification of documents, signature recognition, etc. In this systematic review, we found 11 primary studies that applied CELM in the context of handwritten digit or character recognition.

Table 5 presents the datasets used for handwritten digits recognition found in our systematic review. The two main datasets for digit recognition were MNIST and USPS, and the main dataset used for character recognition was EMNIST. Next, we present a brief description of the two main datasets.

Table 5: Datasets for handwritten digit or character recognition reported in the primary studies.

Dataset	References
MNIST	[24] [34] [74] [75] [27][28] [76] [29] [30] [77]
USPS	[27] [76] [29] [30] [77]
EMNIST	[10]

The Modified National Institute of Standards and Technology (MNIST) [78] dataset contains 70,000 images corresponding to handwritten numeric figures. It is a variation of a more extensive database named NIST, which contains more than 800,000 images with handwritten characters and numbers provided by more than 3,600 writers. The MNIST contains representative images of 10 classes (digits 0 to 9) with dimensions 28 × 28.

The US Postal (USPS) dataset [79] is composed by digital images of approximately 5,000 city names, 5,000 state names, 10,000 ZIP Codes, and 50,000 alphanumeric characters are included. The images have size of 16 × 16.

4.5. Face recognition

Face recognition is commonly present in security systems, tagging people on social networks, etc. It is also common for several machine learning models to use face recognition databases as benchmarking [80]. We found eight works that cover object recognition, about 9% of the primary studies.

Table 6 presents 11 datasets used for face recognition with CELM models found in our systematic review. The YALE dataset was the most used, while ORL was used in two works.

The YALE dataset [85] contains 165 images from 15 different people, with 11 images for each person. Each image contains different expressions such as happy, sad, sleeping, wink, etc.

ORL face dataset [86] is composed of 400 images of size 112 × 92. There are 40 persons, ten images per person. Like the YALE dataset, there are different expressions, lighting setup, and so on.

doi:10.20944/preprints202104.0753.v1

Table 6: Datasets for face recognition reported in the primary studies.

Dataset	References
YALE	[26] [28] [81]
ORL	[29] [26]
Casia-V4	[82]
CMU-PIE	[30]
XM2VTS	[81]
AR	[81]
LFW-a	[81]
FERET	[81]
Youtube-8M	[83]
ChaLearn	[84]

4.6. RGB-D image recognition

RGB-D images are graphical 3D representations of a capture that may be used for object recognition, motion recognition, and so on. In addition to RGB color images, another channel (-D) of information corresponding to depth is added. It is possible to obtain accurate information on the shape and the location of the objects analyzed on the scene. The low-cost Microsoft Kinect sensor is generally used to capture scenarios and objects. With that, several machine learning models are currently used for object recognition [87], human motion [88], among other applications using data from RGB-D sensors [89]. Seven works apply CELM models for the learning process in RGB-D data, representing 8% of the primary studies.

Table 7 presents the datasets used for RGB-D image recognition found in our systematic review. The Washington RGB-D object is the most used dataset. Simultaneously, the other databases are used by only one work: 2D3D object, Sun RGB-D Object, NYU Indoor Scene, Princeton ModelNet, ShapeNet Core 55, Princeton Shape Benchmark, MIVIA action, NOTOPS, and SUB Kinect interaction. Next, we present a brief description of the main dataset, the Washington RGB-D.

Table 7: Datasets for RGB-D image recognition reported in the primary studies.

Dataset	References
Washington RGB-D object	[90] [91] [92] [93] [94] [95]
2D3D object	[94]
Sun RGB-D object	[94]
NYU indoor scene	[94]
Princeton ModelNet	[96]
ShapeNet core 55	[96]
Princeton shape benchmark	[96]
MIVIA action	[97]
NOTOPS	[97]
SBU Kinect interaction	[97]

The Washington RGB-D Object dataset [98] contains 300 objects captured by a Kinect camera with 640 × 480 resolution. Objects are organized into 51 categories. The captures are sequential, that is, records in 3 video sequences for each object recorded.

4.7. Other application scenarios

We also reported works involving scenarios with fewer applications, such as street applications, factories, food classification, textures, documents, etc. Table 8 summarizes the complete list of other applications found in this systematic review's primary studies.

Table 8: Other application scenarios found in the primary studies.

Application	References
Food classification	[99] [100] [31] [35]
Street applications	[101] [102] [103]
Factory	[104] [105] [106]
Motion recognition	[107] [108] [109]
Detection	[24] [110]
Texture classification	[26] [111]
Image Segmentation	[23]
Document recognition	[112]
Criminal investigation	[113]
Animal classification	[35]
Robotics	[114]
Fire detection	[115]
Clothes classification	[116]

5. RQ 2: How are defined the CELM architectures in the analyzed works?

From the primary studies, one can define two main categories of CELM usage: (i) works that use CNN for feature extraction and ELM for fast learning on extracted features, and (ii) works that use ELM for fast training CNN architectures. Both approaches can make better training time and multimedia data learning tasks. Fig. 4 illustrates a summarization of how CELM's are being used in the current literature.

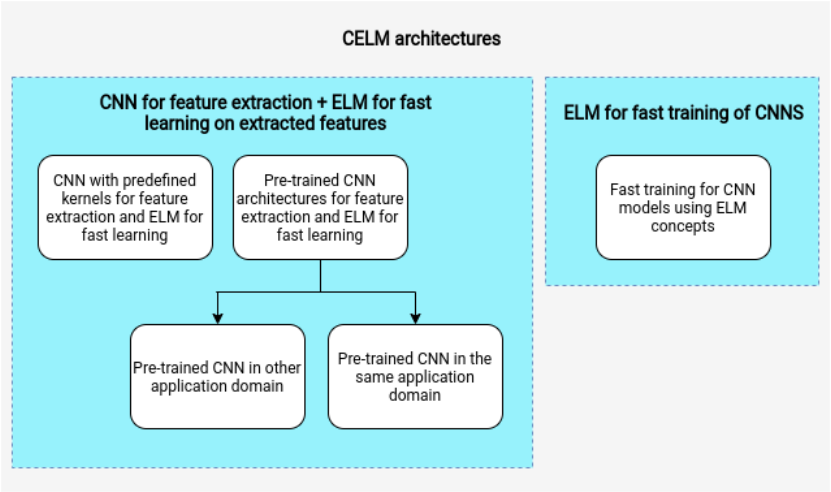
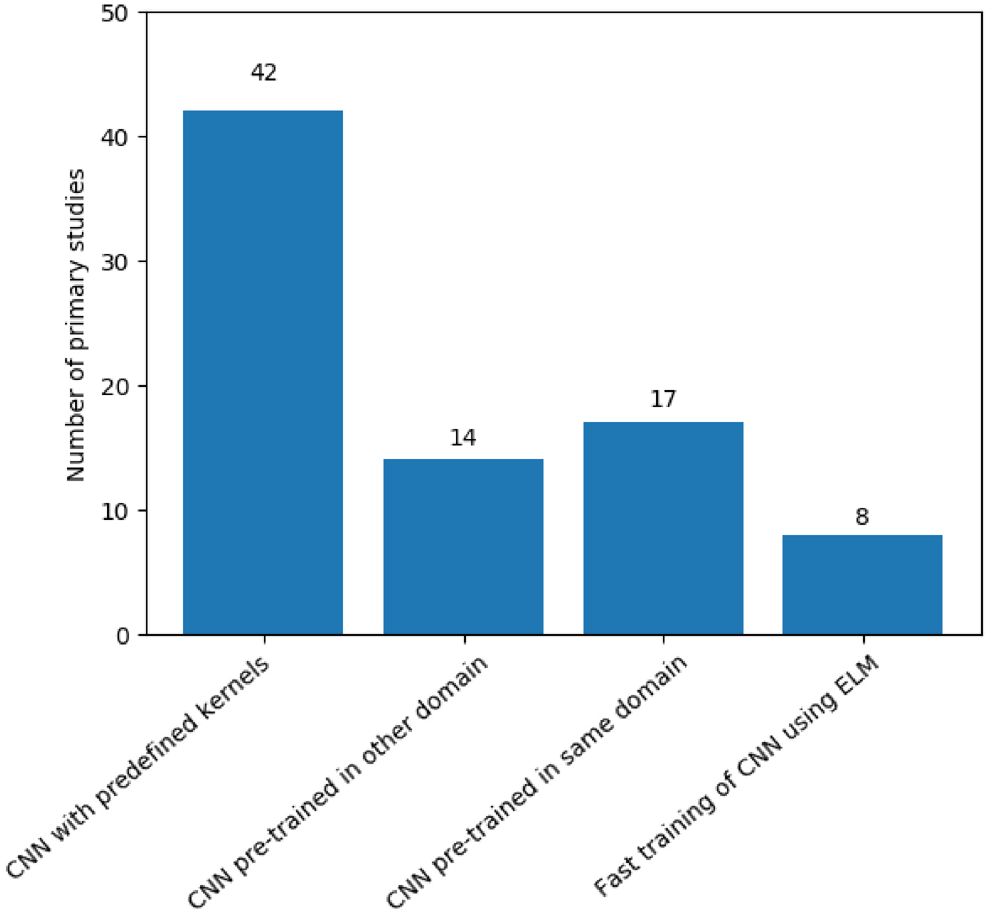

CELM architectures
CNN for feature extraction + ELM for fast learning on extracted features
CNN with predefined kernels for feature extraction and ELM for fast learning
Pre-trained CNN architectures for feature extraction and ELM for fast learning
Pre-trained CNN in other application domain
Pre-trained CNN in the same application domain
ELM for fast training of CNNS
Fast training for CNN models using ELM concepts

Figure 4. Full representation of answers from RQ 2.

In the CNN for feature extraction + ELM for fast learning on extracted features, the power of representation of the inputs increases with deep features, and ELM enables a shorter training time and a high capacity for generalization [6]. There are some ramifications for this approach: (i) CNN for feature extraction using predefined kernel weights (or filters), in which classic image and signal processing filters are used (see sub-section 5.1); and (ii) CNN for feature extraction using previously pre-trained weights, in which the pre-trained weights can be learned in the same or different application domains (see sub-sections 5.2 and 5.3).

The usage of ELM concepts for fast training of classical CNN architectures proposes using a complete CNN, substituting the training process (see sub-section 5.4). The training is no longer done by backpropagation but by algorithms based on ELM to learn the features. This change provides a better training time for CNN and proposes further improvements.

Fig. 5 presents the amount of work per type of CELM usage. More than half of the primary studies use CNN with pre-defined filters to extract features and ELM for training the features (about 54%). The other three types of CELM architectures are distributed in similar quantities. In the following subsections, we discuss how the primary studies applied these different CELM architectures.



0.8

Figure 5. Quantification of kind of works grouped by answers from RQ 2.

5.1. CNN with predefined kernels for feature extraction and ELM for fast learning

This approach is the broadest and most varied when compared to the others. This amplitude occurs because there are many architectures based on default kernel (or filter) initialization. In this approach, CNNs are used as feature extractors without any prior training and the fully connected layer. CNN kernels are pre-defined through processing, statistical distribution or decomposition of values, whereas ELMs or their variations replace fully connected layers. In this approach, the architecture frees backpropagation training and makes the learning process more simple. Fig. 6 shows an generic example of this architecture.

Several kernels can be used in the convolution layers, such as: Random, Gabor, PCA, Patch, and even a combination of these. Also, some works propose techniques for pre and post-processing of the convolutional layers' features. Table 9 summarizes the works that use CELM with pre-defined kernels find the the primary studies. Note that some works fit on more than one approach.

5.1.1. Random filter

The most used kernel found in the primary studies was the kernel randomly generated through a Gaussian distribution, or the random filter: [33], [75], [76], [92], [109], [62], [97], [72], [56], [47], and [99].

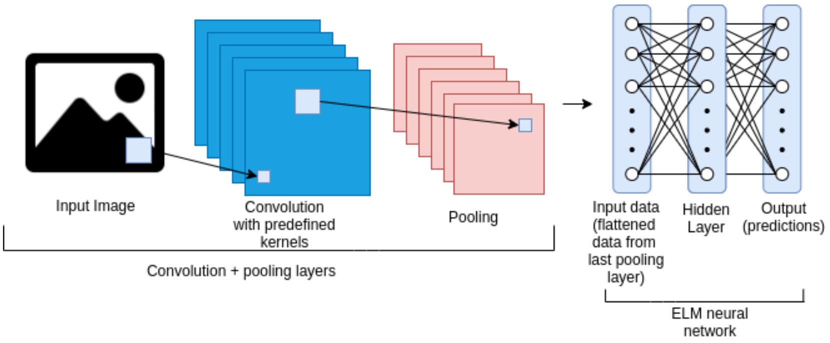

Input Image
Convolution with predefined kernels
Pooling
Convolution + pooling layers
Input data (flattened data from last pooling layer)
Hidden Layer
Output (predictions)
ELM neural network

Figure 6. Example of a CELM architecture composed of a CNN with pre-defined kernels for feature extraction and an ELM for fast learning. The convolution and pooling layers' quantities can be varied. Also, the kernels pre-defined in the convolution layers can be based on different distributions. At the end of the feature extraction process, an ELM network makes the learning process.

Table 9: Variations of works using CELM with pre-defined kernels. *Note that there are works based on CNN with random filters + ELM, which we consider as a special case named ELM-LRF, and therefore are referred specifically in Table 10.

Approaches	**References**
CNN with random filters + ELM*	[33] [75] [76] [92] [109] [62] [97] [72] [56] [47] [99]
CNN with gabor filters + ELM	[104] [48]
CNN with a combination of filters + ELM	[10] [107]
Ensemble of CNN with predefined filters + ELM	[75] [55] [72]
Combination of image processing techniques + CNN with predefined filters + ELM	[92] [55] [23] [117]
PCANet + ELM	[71] [81]

There is a particular case of CELM where the CNN has random filters, but orthogonalized, known as local receptive fields based extreme learning machine (ELM-LRF). ELM-LRF was proposed by Huang et al. [6], and it is based on the premise that ELM networks can adapt themselves and have good generalization when having random features that represent local regions of the input images. The use of the LRF term comes from CNNs, as they can represent different regions of a given image through their convolutions. The network structure consists of a convolutional layer, followed by pooling, while an ELM network is responsible for the training and classification of the extracted features. The convolution kernels are orthogonalized, employing decomposition by singular values (SVD). The convolutional layer applies random filters to extract LRF. Square-root pooling is applied to reduce the dimensionality of the data. Finally, all the traditional learning is done through the ELM network to calculate the inverse Moore-Penrose matrix for training the features generated by the LRF. There is no hidden layer with random weights in the classifier, only one layer of output weights.

Several works applied ELM-LRF in its default form for their learning process [6], [90], [25], [63], [39], [53], and [52]; and some other variations of ELM-LRF, as shown in Table 10.

Table 10: Variations for ELM-LRF reported in the primary studies.

Approach	**References**
ELM-LRF (default)	[6] [90] [25] [63] [39] [53] [52]
Multimodal ELM-LRF	[93] [55] [91] [105]
Multiple kernel ELM-LRF	[26]
Multilayer ELM-LRF	[67] [27] [38] [51] [114] [77]
Autoencoding ELM-LRF	[27] [28] [93]
Multiscale ELM-LRF	[67] [111] [106]
Recursive ELM-LRF	[29]

Some works consider using multiple data sources for parallel feature extraction with ELM-LRF for making a unique final decision. These approaches that consider multiple data sources are named as multimodal [93], [55], [91], and [105].

We presented previously some works that combine different filters in CNNs for feature extraction. This feature combination approach is also used in ELM-LRF architecture, multiple kernel ELM-LRF [26]. In this work, the authors propose using a variation of Gabor filters with random filters, and for this reason, the authors name this approach ELM-hybrid LRF (HLRF). The authors carry out experiments to define the p and q values of the Gabor filters and perform an analysis of the number of layers that provide optimal accuracy values.

The multilayer ELM-LRF is another known ELM-LRF variation which consists of multiple convolution and pooling layers [67], [27], [38], [51], [114], and [77].

Autoencoding ELM-LRF proposes high-level feature representation using ELM-AE with ELM-LRF and is proposed by [27], [28], and [93]. Another notable difference is using three ELM-AE used in parallel for each respective colour channel for coding features. The work [93] proposed a Joint Deep Random Random Convolution and ELM (JDRKC-ELM) model for recognition of two data modality separately (application of ELM-LRF). After feature extraction, the fusion layer with coefficient to combine two features type and ELM-AE learn top-level resource representations. ELM classifier is responsible for the final decision.

Also, some works consider all channels or variate scales (multiscale) of the images by applying different ELM-LRF architectures for feature extraction, and learning task [67], [111], [106], and [54].

Furthermore, finishing the ELM-LRF variations, the work [29] presents two a recursive model based on ELM Random Recursive Constrained (R2CELM) and ELM based on Random Recursive LRF (R2ELM-LRF), which are constructed by stacking CELM and ELM-LRF, respectively. Following the concept of stacking generalization, random projection and kernelization are incorporated in the proposed architectures. R2CELM and R2ELM-LRF not only fully inherit the merits of ELM but also take advantage of the superiority of CELM and ELM-LRF in the field of image recognition, respectively. R2CELM and R2ELM-LRF demonstrate their best performance in precision tests on the six sets of reference image recognition data by empirical results.

5.1.2. Gabor filter

The works [104], and [48] use the Gabor filter, which is considered similar to the human visual system and is widely used in general computer vision tasks, not only in CNNs. The Gabor filter is linear, and it is generally used for analyzing textures in images. The frequency and orientation attributes represent the Gabor filters. Similar to the random filter, the Gabor filter used in [104] and [48] obtained a high capacity for representing the data and could provide a better generalization of ELM.

5.1.3. CNN with combination filters

Other studies use a combination of different filters in the convolutional layers. In the work [10], the authors observe that CELM approaches in the literature have the limitation of using random filters in only one step of extracting features. Due to random filtering usage limitation, the authors propose combining the following filters: random filter, patch filter (sub-regions was chosen from an input image), principal component analysis (PCA) filters and Gabor filter. In [107], authors apply the Gabor filter with different values of directions and scales in the first convolutional layer, and the Radial Basis Filter is applied in the second convolution layer. After each convolution layers, the data are pooled by pooling layers. Both approaches provide good generalization capacity.

5.1.4. Ensemble of CNN with pre-defined filters

Ensemble approaches of CNNs and ELMs have been considered by [75], [55] and [72]. An ensemble generally consists of combining more than one learning model for a final decision [118]. Fig. 7 illustrates an example of ensemble of CELM.

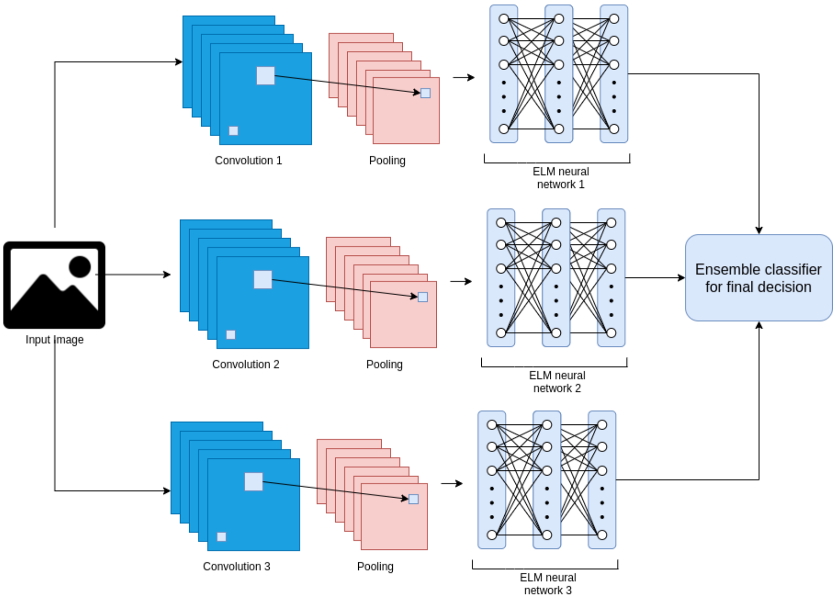


Figure 7. Example of ensemble representation for CELM. Three different CELM architectures are used for feature extraction and learning process. At the end, an operator is responsible to make the final classification decision.

In [75], authors use three CELM architectures combined through the majority voting ensemble. Each sub-architecture consists of three convolutional layers, each one followed by a pooling layer, then at the end, an ELM is responsible for the training and classification process. In [72], authors train three different ELM networks for the classification process. Each ELM network has as input the last two convolutional and last pooling layer, and an ensemble makes the final decision of these three ELMs. The work presented in [55] was described previously in multimodal ELM-LRF.

5.1.5. Combination of image processing techniques and CNN with pre-defined filters

In addition to convolutions, pooling and ELM, some works also consider image processing techniques for pre and post-processing of images or features: [92], [55], [23] and [117]. Authors in [92] propose using K-means in the inputs, then convolution filters are applied. Spatial Pyramid Pooling and a recursive neural network are applied to the abstraction of the data generated before applying the ELM for training and classification. In the end, the ELM is used for feature learning and classification. The approach proposed in [55] consists of the feature extraction by CNN in two types of input: (i) the original image and (ii) image after transformation of rotation through fractal extraction and segmentation. After that, the features are combined and trained by two ELM networks. The final decision is made by combining the outputs of these ELMs. In [23], authors propose the extraction of superpixels using the Simple Linear Iterative Clustering (SLIC) algorithm. With that, the extraction of candidate regions with their corresponding labels is done. The CNN architecture is applied in these candidate regions for feature extraction so that the ELM performs the prediction of semantic segmentation in the images. In [117], the image data is captured, and a search is done for colour similarity in the image. After that, segmentation is applied. Finally, two convolutional layers are applied, each one followed by two pooling layers. With that, the data is classified by a KELM (a KLM with RBF kernel).

5.1.6. PCANet

The work [71] presents a classification approach which consists of using the PCANet [119] network to extract features using the Principal Component Analysis (PCA) algorithm in convolutions in the images. Then, the ELM with the composition of several kernels is used for the classification task. The proposed approach presents promising results. The work [81] develops a new approach to image classification using a new architecture, the 2DPCANet, a variant of PCANet. While the original PCANet network performs 1D transformations for each image line, the 2DPCANet performs 2D transformations in the entire image. As a result, there is a refinement in the process of extracting features. At the end of the feature extraction, the training with the ELM network is carried out. The architecture is evaluated in a different dataset and shows improved accuracy compared to the original architecture.

We observed that all works in this section use small CNNs architectures. Authors usually do not specify how to define the ideal number of layers and filters. When the number of layers and filters is increased, the amount of data to be processed by ELM also increases. In the literature, classic machine learning algorithms tend to perform the learning task with more incredible difficulty when having a vast amount of data. Besides, computational processing time increases in proportion to the complexity of the CNN architecture. For these reasons mentioned above, the works proposed simpler CNN architectures for extracting characteristics since the objective is to obtain maximum accuracy without gradually increasing the computational cost.

5.2. Pre-trained CNN in other application domain for feature extraction and ELM for fast learning

It is necessary to have many data and machines with a tremendous computational capacity to train deep learning models. Machines with dedicated hardware with GPU processing can be used for training such models, but large amounts of data or resources may not be available for the creation of the models. Therefore, the concept of transfer learning was proposed to deal with these problems.

In transfer learning, the knowledge learned to perform one task can perform another task [120]. In this process, the features that one model learned to perform a task can be transferred to another model to perform a different task. A minor adjustment (named fine-tuning) needs to be done on the last layers of the model (usually the fully connected layers) [121]. In this systematic review, we reported works that propose a fine-tuning approach using an ELM-based classifier. This approach is similar to the previous ones reported in section 5.1. The difference is that a pre-trained CNN (generally without the fully connected layers) is used to make the feature extraction process. An ELM-based classifier is used to make a new training process in the extracted features. Note that here we name this process as fine-tuning with ELM. Fig. 8 illustrates the transfer learning process with ELM.

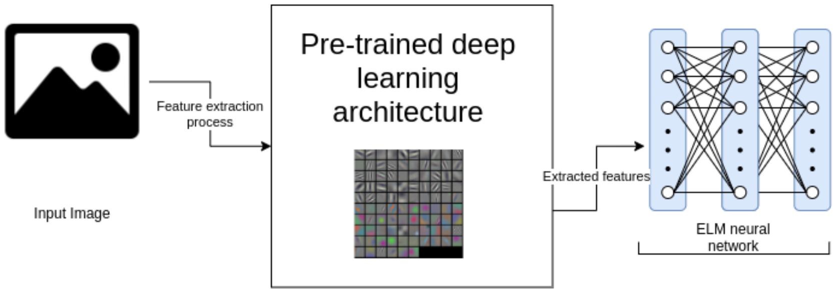

Input Image
Feature extraction process
Pre-trained deep learning architecture
Extracted features
ELM neural network

Figure 8. Example of the usage of pre-trained deep learning architectures and fine tuning with ELM.

We found various works that use classic deep neural architectures for a transfer learning task, suck as: AlexNet, CaffeNet, VGGNet, GoogLeNet, Inception-V3, ResNet, SqueezeNet. Also, own deep pre-trained architectures are proposed by some works for the transfer learning task. Table 11 resumes the works that use pre-trained deep learning architectures and fine-tuning using ELM.

Table 11: Pre-trained architectures used for fine tuning with ELM reported in the primary studies.

Pre-trained architecture	References
AlexNet	[113] [57]
CaffeNet	[69] [101] [122]
VGGNet	[57] [69] [31] [68] [115] [83] [66] [96] [100]
GoogLeNet	[57] [66] [82]
Inception-V3	[31]
ResNet	[31] [115] [66] [96] [100]
SqueezeNet	[61]

The first deep learning architecture used for transfer learning and fine-tuning with ELM that we cover is AlexNet [123]. The AlexNet architecture is one of the pioneers responsible for popularizing deep learning for image recognition. This architecture has five consecutive convolutional layers with filter size equals to 11 and pooling. After each convolutional layer, the Rectified Linear Units (ReLU) activation is used to reduce the classification error. Three fully connected layers are responsible for data classification. AlexNet was initially trained in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) dataset using multiple GPU. The authors adopted the freezing part of the weights (dropout) and data augmentation overcame the overfitting problem. The architecture reached an error of 15.3% in the database used in the year 2012, and it was much higher than other architectures. With all the acquired learning, the architecture is used to extract characteristics, then removing the fully connected layers.

Sone works used the AlexNet architecture with the pre-trained weights in the ILSVRC dataset in conjunction with ELM networks to replace fully connected layers, thus fine-tuning [113], [57]. There is a variation of the AlexNet model using a unique GPU for training task, and the variation is named as CaffeNet [124]. CaffeNet model with pre-trained weights in the ILSVRC dataset was also used for feature extraction and fine-tuning with ELM [69]. Using CaffeNet, the works [101], and [122] presented a new architecture considering the canonical correlation between visual resources and resources based on biological information. The use of discriminative locality preserving canonical correlation analysis (DLPCAA) was adopted after the feature extraction stage, considering the information on the label and preserving the local structures for calculating correlation. In the second layer, training with ELM was performed, which does not need many images for training.

Another important classic deep learning architecture used for fine-tuning with ELM is VGGNet [18]. There are variations of VGGNet as VGG-16 and VGG-19, which contain several characteristics in common such as the number of convolutional layers being varied, all containing three fully connected layers. Firstly, the use of smaller local receptive fields with kernel size equals three stands out, unlike AlexNet, which equals 11. VGGNet architectures have five blocks of convolutional layers with ReLU and pooling, and the number of filters varies from 64 to 512. The architecture has many convolutional layers that can increase the data representation capacity and successfully transfer learning applications.

We reported the use of VGG-16 [31], [68], [115], [69], [57], [66] and VGG-19 [100] architectures for extracting features and fine-tuning with ELM, all previously mentioned works used pre-trained weights from ILSVRC dataset. The work [83] presented an approach to predict and classify data using a multimodal approach, where video data (frame sequencing) and audio are considered. The image data is extracted with the VGG-16, the audio data is processed with LSTM. Finally, the data was trained and classified with an ELM. The authors in [96] presented a computationally efficient method for image recognition. A new structure of multi-view CNNs and ELM-AE has been developed, which uses the composite advantages of VGG-19 deep architecture with the robust representation of ELM-AE features and the fast ELM classifier.

GoogLeNet and Inception-V3 are other pre-trained CNN in the ILSVRC database for feature extraction and fast learning with ELM. GoogLeNet architecture [125] makes use of 1×1 convolutions coupled in named Inceptions modules, which reduces the computational cost. Instead of using fully connected layers, global average pooling is used. This reduction by global average pooling reduces the representation of the feature maps to a single value directly connected with the softmax layer for class prediction. These are the works from this systematic review that use GoogLeNet with pre-trained weights on the ILSVRC dataset to extract features [82], [57], [66]. Inception-V3 is an evolution of the previous model containing regularization, grid size reduction and factorizing convolutions. Inception-V3 [126] is also considered a good alternative for transfer learning and has been considered in the literature for fine-tuning with ELM [31].

Deep residual network (ResNet) [19] was proposed for ILSVRC 2015, which was the winner with a classification error of 3.57%. ResNet contains identity shortcut connections, which are skipping connections in convolutional layer groups. The idea behind skipping connections in ResNet is to prevent the network, which is very deep, from dying due to the gradients evolving. ResNet uses a signal that is the sum of the signal produced by the two previous convolutional layers plus the signal transmitted directly from the point before these layers. We found some works that used ResNet for transfer learning using the pre-trained weights from ILSVRC dataset in conjunction with ELM: [31], [115], [100], [66], [96].

In [127], the deep network SqueezeNet was proposed, which is also used for transfer learning with the neural network ELM. However, this architecture provides the same accuracy as AlexNet in the ILSVRC database, with fewer trained parameters. The data is compressed and processed in Squeeze layers, which are composed of 1x1 convolutions. The data expansion is done through more convolutional layers 1x1 and 3x3, expanding the local receptive fields. As a result, the architecture is simpler to process and provides a good representation of the data. The work [61] used SqueezeNet for feature extraction and ELM for training the extracted features. Also, the approach used the Super Resolution Fuzzy-C-Means (SR-FCM) clustering technique for image segmentation.

5.3. Pre-trained CNN in same application domain for feature extraction and ELM for fast learning

We previously discussed deep and transfer learning models applied to feature extraction and fine-tuning with ELM networks. These transfer learning models previously reported are pre-trained in another application domain. This systematic review also reports papers that use pre-trained transfer learning architectures in the same application domain. The objective then is not to decrease the computational cost (when the authors train the architectures) but rather to increase the proposed final architecture's accuracy.

There are two types of application of the approach discussed in this section. Firstly, some works have been trained in the ILSVRC database and are used for fine-tuning in the same application domain as the ILSVRC. Secondly, some works fully train architectures (classic or own) and then immediately use the network to extract features in the same work for later fine-tuning with ELM. We will discuss these two types of application below. Table 12 refers to transfer learning architectures used in this approach.

Table 12: Pre-trained architectures used for fine tuning in the same domain with ELM reported in the primary studies.

Pre-trained architecture	**References**
AlexNet	[112] [37] [36] [32]
VGGNet	[94] [84]
MobileNet	[108]
DenseNet	[35]
Own architectures	[46] [64] [102] [49] [116] [103] [65] [70] [50]

Authors in [112] propose an approach for image classification based on a hybrid architecture of CNN and ELM. The CNN architecture used is AlexNet with pre-trained weights from the ILSVRC dataset (same current application domain). The work performs two stages of training. The first training stage consists of re-training the model in the same work's database, using the complete AlexNet architecture. The second training stage consists of the usage of the trained architecture as a feature extractor. The dense layers are removed, and an ELM network replaces it, performing fine-tuning on the feature vector. As expected, the training time gains in performance in the test.

The authors in [37] and [36] propose an approach to object recognition using a hybrid approach, in which the AlexNet architecture is responsible for the training and the feature extraction. Fine-tuning is done with training sets from different proposed datasets. With its variants like adaptive ELM (AELM) and KELM, ELM is used in the data classification stage. The KELM provides the best accuracy, higher than ELM, SVM, and MLP.

The work [32] combines AlexNet and ELM for image classification in robots' artificial intelligence. CNN is used for feature extraction. As a result, the CNN and ELM classifiers' use shows a faster learning rate and an accurate classification rate than other classification methods. The ReLU activation function is used on the ELM network, obtaining better performance than the existing classification methods.

The work [94] presents an approach for image classification in the scene that is invariable from the camera's point. The authors use a pre-trained convolutional neural network (VGG-f) to extract features from different image channels. VGG-f is another variation of the VGGNet architecture. To further improve performance, the authors created the HP-CNN-T, an invariant descriptor. The convolutional hypercube pyramid (HP-CNN) may represent data at various scales. The classification results suggest that the proposed approach presents a better generalization performance. In [84], the authors present a multimodal approach for regression. The approach consists of feature extraction from using the VGG-19 and VGG-face networks. Data were merged and trained with the KELM network for regression (since a probability-based estimation is made).

To compact deep learning models, the MobileNet model [128] was developed to be small and adaptable for mobile devices and less processing power. Every standard convolution is factored into a depth-to-depth and pointwise 1 × 1 convolution. Authors in [108] use the MobileNet model in a multimodal approach. That is, it receives data from three different types of data for the training process. Thus, the authors perform the training process on the MobileNet network for each data sources. The re-trained MobileNet networks are used as feature extractors through each network's last fully connected layer. Each set of features extracted by the different data sources are trained in three different KELM networks. Finally, the results generated by each KELM are combined through an ensemble-based decision rule.

DenseNet [129] is another classic deep learning architecture. Each convolutional layer of the network receives additional input from all previous layers and passes its feature maps and all subsequent layers. Unlike ResNet, where concatenation is in blocks via gates, each layer receives information from all previous layers. The work [35] presents an approach to image classification using a DenseNet for training and feature extraction and KELM for fine-tuning. The authors perform training of the DenseNet deep network in the proposed dataset. After that, the trained DenseNet is used for feature extraction. Finally, the approach uses a KELM to train the features extracted.

Different from the previous work, other authors propose their architecture instead of using a known network for the transfer learning, such as [49], [46], [64], [102], [103], [116] that propose CELM architectures with a different number of convolutional and pooling layers. The authors use CNN architectures for training the data with the fully connected layers. After the training, authors use their trained networks for feature extraction. They then use features extracted in the ELM network (or its variants) for a new training process and later data classification.

The authors in [65] present the Perceptual Hash-based Convolutional Extreme Learning Machine (PH-C-ELM) to classify images using a three-stage hybrid. This architecture uses a convolutional network in the data generated by Discrete Wavelet Transform-Singular Value Decomposition (DWT-SVD) values after the feature extraction step for data subsampling. The authors present a fine-tuning approach, where the proposed CNN is trained in the data, and then it is used as a feature extractor. Finally, an ELM is trained in the extracted features.

The work [70] presents an approach for image classification in multidimensional sliced images. The authors proposed five different CNN3D architectures (each input consisted of 20 slices per multidimensional image). The training process is conducted by fully connected layers (softmax). Each CNN architecture produces different local receptive fields, and therefore different features. After the CNN training, the architectures are used as feature extractors, and then the features are combined for new training in an ELM.

The authors in [50] present an architecture for image classification, which employs convolution-deconvolution layers and an optimized ELM. Deconvolution layers are used to enhance deep features to overcome the loss of information during convolution. A full multilayer CNN is developed, consisting of convolution, pooling, deconvolution layers, ReLU, and backpropagation. Also, the PCA algorithm is used to extract the first principal component as a training tag. The deconvolution layer can generate enlarged and dense maps, which extract high-level refined resources. The results demonstrate that the proposed structure surpasses other traditional classifiers and algorithms based on deep learning. This is the unique work of the systematic review to use deconvolution layers.

5.4. *Fast training of CNNs using ELM concepts*

Unlike the other aspects presented so far, such as typical CNN for feature extraction + ELM for training the extracted data, there are also approaches that consist of using a complete training of CNNs using ELM concepts. The learning process is not based on the use of the backpropagation algorithm. ELM concepts are used to calculate the error and update the filters and weights based on the pseudo inverse Moore-Penrose. This ensures fast and efficient training, in addition to offering better data representation and generalization capabilities. Next, we present the works that use ELM concepts for fast training.

Authors in [34] use an approach to the representation of features based on the PCANet network and ELM autoencoder. The proposed architecture aims to understand and extract features for the most diverse applications with low computational cost. Three main stages carry out the learning process: (i) obtaining filters and weights with ELM autoencoder and ELM decoder with convolutional layers; (ii) usage of max-pooling operation to reduce the dimensionality of the data; and (iii) post-processing operations such as binary hashing and block-wise histogram, to combine the features obtained to be used in the final classification step. The authors suggest that any classifier can be used to learn the features obtained. The error results in comparison with PCANet shows that the proposed model has a lower error rate in all evaluated scenarios, in addition to offering fast training using ELM neural network.

The work [74] proposes a convolutional neural network model with training inspired by ELM. The convolutional network consists of only two layers: convolutional and pooling, disregarding the fully connected layers. A convolutional layer replaces the fully connected layers with a 1x1 kernel, similar to the GoogLeNet. The steps for modelling and training the proposed network are followed by applying convolution filters in all image regions, forming $n \times n$ window matrices. A reshape is applied to each window, and the filters are learned with an ELM-based approach. This approach provides calculating the More-Penrose pseudo-inverse matrix and updating weights and biases of the convolutional layers. The authors compare the proposed approach with a typical CNN with the implemented backpropagation. Although the proposed approach obtains slightly less accuracy than the baseline, it is worth considering that the training time is 16 times longer than the baseline, indicating that it is possible to obtain high accuracy with little training time.

Authors in [24] propose a new network named CNN-ELM for classification and detection of objects in images, applying the ELM concept at two levels. The first level uses ELM for training the convolutional layers. In these layers, random filters are applied together with the ELM-AE to improve these kernels through autoencoders' representation. In the second level, the extracted features are classified with the multilevel ELM (ML-ELM), an ELM neural network with multiple layers, following the concepts of deep learning. The use of this architecture provides fast processing, however, at a high memory cost. Due to this problem, the authors propose to use batches (or blocks) of data to be trained in memory. In comparison with several baseline architectures, the proposed model obtains the best accuracy and training time.

The work [130] proposes a new architecture and a training algorithm for convolutional neural networks. The Network in Network (NIN) and ELM architecture combined with CNN is adopted, exploring each one's advantages. This architecture naturally exploits the efficiency of extracting random and unsupervised resources, consisting of a deeper network. The images' input is converted into localized patches that are vectorized. They are divided into clusters to pass through the Parts Detector (ELM), where random weights adjust the hidden layers. They are submitted to ELMConv, where random convolutional filters with a sigmoid activation function are used, returning unsupervised convolutional filters. They pass through the ReLU activation function, and an average grouping, normalization and final classification are performed with the ELM.

In [110], authors propose a new approach for performing object tracking using convolutional networks with a modification in the training model. The proposed CNN architecture contains two convolutional layers followed by two layers of poolings, and there are also the traditional fully connected layers with a softmax activation function. The authors still use the descending gradient to update the network's weights and filters with a modification. An autoencoder ELM is used to learn and update the layer weights between the first pooling layer and the second convolutional layer. This provides a reduction in training time and, consequently, a gain in performance.

The work [30] proposes a new architecture named ELMAENet for image classification. The proposed architecture includes three layers: (i) convolutional layer with filter learning through ELM-AE; (ii) non-linear processing layer, where the values are binary with hashing and histogram; (iii) pooling layer. The learning of these features is in charge of the ELM-AE structure. The architecture is evaluated using several datasets and compared with several models, achieving the best computational performance.

Authors in [131] propose an approach for image classification using CNN and ELM. The work's main contribution is regarding a new method of extracting features, where convolutional layers are used with learning filters without the need for the backpropagation algorithm. The authors use ELM-AE to learn the best features in the convolutional layers. An ELM ensemble is used for the data classification. The proposed architecture is evaluated using different datasets, and in three of them, it obtains the best result in terms of accuracy.

The work [132] presents a new approach to train CNNs using ELM and applies it for image recognition. The architecture consists of three convolutional and two pooling layers. There are ELM networks between the two pooling layers and the subsequent convolution layers. Also, there is an ELM network to carry out the recognition stage of the tracks. The error is propagated from the last ELM network about the target (labelled image) in the opposite direction of the network until the first convolution layer is reached. From that, convolution weights, filters and other parameters are adjusted with the intermediate ELM networks, providing a faster adjustment and learning. The proposed approach is superior to others in the literature in terms of accuracy and computational performance.

6. RQ 3: Which are the main findings when applying CELM in problems based on image analysis?

Based on the scenarios and the most common datasets used in the primary studies, in this subsection, we describe the main findings when applying CELM in image analysis.

Next, we present the accuracy results of the CELM models using the primary databases presented in section RQ1. We also present the time required for training and testing the CELM models. It is worth mentioning that the presentation of these time results emphasizes that CELM models are trained and tested in less time than classic machine learning models and not to compare them against each other, as each model was trained and tested in different machines with different setups.

Authors in [6], [25], [24], [26], [27], [28], [29], and [30] applied CELM to solve the object recognition problem, using the NORB dataset. In general, all works presented a good accuracy, all of them over 94%, as shown in Table 13. All these works performed comparisons against algorithms such as classic CNN, MLP, SVM, and the CELM models outperformed all of them.

The best accuracy results were 98.53%, and 98.28%, found by [28] using an ELM-LRF with autoencoding receptive fields (ELM-ARF) and [30] using an ELMAENet, respectively. It demonstrates the excellent representativeness of the extracted features and generalization capability of ELM models.

In general, we noted that some works just presented the training time in the papers that considered the NORB dataset in their experiments. For this reason, in Table 13, we do not consider testing time in the discussion. The best training time was achieved by [27] (216 seconds), and this result was probably due to the compact autoencoding features by ELM-AE. The worst result was achieved by [29] (4401.07 seconds). The difference can probably be due to the different machine and scenario setup, as previously discussed. In general, ELM-LRF-based architectures provide a low training time due to the simplicity of that architectures. All these architectures presented better training results than classic machine learning models.

Table 13: Results obtained by CELM architectures for object recognition in the NORB dataset.

Reference	Approach	Accuracy	Training time (s)
Huang et al. (2014) [6]	ELM-LRF	97.26	394.16
Bai et al. (2015) [25]	ELM-LRF	97.24	400.78
Yoo and Oh (2016) [24]	CNN-AE-ML-ELM	94.92	1165.87
He et al. (2019) [26]	ELM-HLRF	97.45	516.08
Wu et al. (2020) [27]	ELM-ARF	98.00	216
Wu et al. (2020) [28]	ELM-MAERF	98.53	279
Song et al. (2020) [29]	R_2ELM-LRF	97.61	4401.07
Chang et al. (2020) [30]	ELMAENet	98.28	-

Authors in [51], [48], [54], [50], [49], [46], and [47] used the Pavia dataset for remote sensing classification. Note that remote sensing approaches use another evaluation metrics such as average accuracy (AA), overall accuracy (OA), and Kappa, as shown in Table 14.

The most common approach used to this end is a CNN previously pre-trained in the Pavia dataset used for feature extraction and an ELM for classification task [50], [49], [46]. However, the ELM-HLRF proposed in [51] achieved the best AA and OA results, respectively 98.25% and 98.36%.

Most works did not report any results regarding the training or testing time, but we show the effectiveness in these metrics for remote sensing classification. The work [46] reported a low training time, achieving 14.10 seconds, and the work [47] reached 0.79 seconds of testing time.

Table 15 presents accuracy results using the MNIST dataset for handwritten digit recognition done by [24], [77], [34], [76], [74], [75], [29], [30], [27], and [28]. All works

Table 14: Results obtained by CELM architectures for remote sensing classification in the Pavia dataset.

Reference	Approach	AA	OA	Kappa	Training time (s)	Testing time (s)
Lv et al. (2016) [51]	ELM-HLRF	98.25	98.36	0.981	44.12	-
Shi and Ku (2017) [48]	CNN(gabor)-ELM	94.3	92.8	0.940	-	-
Shen et al. (2017) [54]	ELM-LRF	97.95	98.29	0.981	-	-
Li et al. (2018) [50]	CNN(pre-trained)-ELM	-	96.70	0.955	-	-
Cao et al. (2018) [49]	CNN(pre-trained)-ELM	97.50	98.85	0.983	-	-
Huang et al. (2019) [46]	CNN(pre-trained)-ELM	85.50	87.77	0.860	14.10	25.24
Shen et al. (2019) [47]	CNN(random)-ELM	-	97.42	0.971	49.00	0.79

presented a high accuracy, over 96%. The training time varied considerably, ranging from 8.22 seconds [76] to 2658.36 seconds [29]. Regarding to the testing time, the work done by [76] also presented the best performance (0.89 seconds).

Different neural network implementations can make a difference in processing time, which can explain the difference in the work done by [76] to others. Besides having the best training and testing time, the work [76] achieved the worst accuracy for the handwritten digit classification task (96.80%).

We highlight the work presented in [30], which outperformed other accuracy metric models (99.46%) using an ELMAENet. The results showed that feature representation in ELM-LRF and CNN with ELM-AE was sufficient to reach a good accuracy result. In the learning task, the accuracy got superior to 99% in both cases. The results obtained by the works demonstrate that CELM approaches have good generalization performance in this benchmark dataset.

Table 15: Results obtained by CELM architectures for handwritten digit recognition in the MNIST dataset.

Reference	Approach	Accuracy	Training time (s)	Testing time (s)
Yoo and Oh (2016) [24]	CNN-ML-ELM-AE	99.35	1113.09	-
Pang and Yang (2016) [77]	ELM-HLRF	98.43	27.8	-
Cui et al. (2017) [34]	PCANet-ELM-AE	99.02	-	-
Ding et al. (2017) [76]	CNN(random)-ELM	96.80	8.22	0.89
Khellal et al. (2018) [74]	ELM-CNN	99.16	157.08	-
Kannojia and Jaiswal (2018) [75]	CNN(random)-ELM	99.33	-	-
Song et al. (2020) [29]	R_2ELM-LRF	99.21	2658.36	-
Chang et al. (2020) [30]	ELMAENet	99.46	-	-
Wu et al. (2020) [27]	ELM-ARF	98.95	265	22
Wu et al. (2020) [28]	ELM-MAERF	99.43	204	14.8

Table 16 shows the results related to the YALE dataset for face recognition obtained by the works [81], [26], and [28]. All works reported an accuracy superior to 95%. The best accuracy result was found by [81] (98.67%), and the worst was reached by [26] (95.56%).

The accuracy result obtained by [81] (PCA convolution filters) and [28] (multiple autoencoding ELM-LRF) demonstrate that the use of multiple random or Gabor filters was not sufficient to provide good representativeness of the data for training in ELM using the YALE dataset. The works ([81] and [28]) have more robust architectures, which can explain the better accuracy result.

Only the work [28] presented training and testing times, being 16 and 0.38 seconds, respectively. The literature suggests that CELM approaches can also reach good accuracy results in the face recognition problem. On the other hand, the training and testing time was not clear due to the missing reported results.

Table 16: Results obtained by CELM architectures for face recognition in YALE dataset.

Reference	Approach	Accuracy	Training time (s)	Testing time (s)
Yu and Wu (2018) [81]	2DPCANet-ELM	98.87	-	-
He et al. (2019) [26]	ELM-HKLRF	95.56	-	-
Wu et al. (2020) [28]	ELM-MAERF	98.67	16	0.38

doi:10.20944/preprints202104.0753.v1

Table 17 shows the results when solving RGB-D image recognition using the Washington RGB-D Object dataset by the works [90], [91], [93], [92], and [94]. RGB-D image recognition is a task that considers two types of data, such as the RGB color channel and the depth, which makes the classification task harder. The accuracy results varied from 70.08% (single ELM-LRF) to 91.10% (VGGNet-ELM). One can note improvements when the RGB and D channels are separately processed in random filter representations [91], [93], [92]. There is no significant difference in the results reached in feature extraction by random convolutional architectures (up to 90.80% [92]) and pre-trained architectures (91.10% [94]). Besides the high complexity for RGB-D classification, the CELM architectures reached good accuracy. Besides provided a low accuracy, the work presented in [90] reached the best training time due to its network complexity (192.51 seconds).

Table 17: Results obtained by CELM architectures for RGB-D classification in Washington RGB-D Object dataset.

Reference	Approach	Accuracy	Training time (s)	Testing time (s)
Boubou et al. (2017) [90]	ELM-LRF	70.08	193.51	0.645
Liu et al. (2018) [91]	MMELM-LRF	89.30	715.66	-
Yin and Li (2018) [93]	JDRKC-ELM	90.80	615.32	-
Yin and Li (2019) [92]	CSPMPR-ELM	90.80	-	-
Zaki et al. (2019) [94]	VGGNet-ELM	91.10	-	-

In general, one can note that the CELM models provide satisfactory results in terms of accuracy and computational performance (training time and testing).

CNN-based approaches with predefined kernels for feature extraction provide good results in terms of accuracy and training time. In two scenarios (object and face recognition), architectures of this type presented better accuracy ([27], [81]) than others approaches, such as deep belief network and stacked autoencoders. The excellent performance of this approach in the computational aspect is due to its one-way training style. The feature extraction is the most costly stage due to the high number of matrix operations in the CNN. However, when it comes to the training stage using ELM, the processing time is not an aggravating factor, except when the architectures' complexity is increased.

Regarding the approaches that use pre-trained CNN architectures (same or other domain) to extract characteristics and later fine-tuning with ELM, it is also observed that the results are satisfactory. This approach outperforms others in the remote sensing, and RGB-D image recognition scenarios [49], [94] considering the accuracy metric. Classic CNNs and support vector machines are examples of outperformed approaches. This approach's training method is also a one-way training style, which explains the excellent training time involved in the learning process.

The fast training approaches for CNN models using ELM concepts could not be further analyzed because only a few works were found in the literature. However, one can note that such an approach outperforms other CELM models such as ELM-LRF and PCANet-ELM in terms of accuracy when considering the handwritten digit recognition problem [30]. Instead of using the backpropagation algorithm for feature training, the authors used the ELM-AE network, obtaining a more compact representation of data and better training time.

In general, CELM presented interesting results regarding the accuracy compared to several proposals found in the literature. Despite not having the same power as the conventional CNNs (with fully connected layers and backpropagation) to extract features, CELM's accuracy proved competitive in the analyzed scenarios and benchmark datasets. The competitiveness of the results is clear when, in many cases, CELM was superior to several traditional models such as MLP (as in [102], [69], [50]) e SVM (as in [46], [113], [90]). Observing these results, we reported a good generalization and good representativeness by CELM [104], [68], [97], [27], [55], [57], [49].

From the primary studies, we also notice that CELM architectures have good convergence and provide better accuracy. Changing the fully connected layers to ELM network consequently increase the training speed and avoid fine adjustments [115], [122], [30], [131]. Convergence is achieved without iterations and intensive updating of the network parameters. In the case of CNN for feature extraction + ELM, the training is done with ELM after the extraction of CNN features. Rapid training reflects directly on computational performance. With the adoption of CELM, it is possible to decrease the processing time required for the learning process. This feature makes CELM able to solve problems on a large scale, such as real-time or big data applications [94], [111], [29].

7. RQ 4: What are the main open challenges in applying CELM to problems based on image analysis?

Despite the many advantages of the CELM architectures, such as suitable training time, test time, and accuracy, some open challenges can serve as inspiration for future researches, contributing to the advancement in the field of research in CELM.

It is known that the number of layers can be an important factor in the ability to generalize a neural network. Classic works of deep learning proposed architectures with multiple convolution layers [18], [19], [129]. However, when the number of layers is increased, problems with increasing training time and loss of generalization capacity emerge [77], which can cause overfitting issues. These two reasons may explain that many CELM architectures with predefined kernels do not use very complex architectures to extract features.

Despite GPUs' good performance, sometimes it is not possible to use them in the real environment. When this happens, all data is stored sequentially in the RAM and processed by the CPU, increasing the training time, especially when handling data with high dimensionality. One possibility to overcome this issue is using approaches that aim at high-performance computing using parallel computing. Also, the usage of strategies for batching the features can replace the number of samples N in the memory requirements [9]. There is an approach in the literature that aims to use ELM for large-scale data problems known as high-performance extreme learning machine [9], which could be adequately analyzed in the context of CELM.

Regarding the problem of the number of convolutional layers, the gradual increase in the complexity of the network can cause problems in the model generalization. It can decrease the accuracy and also cause overfitting. Some works in the literature have proposed using new structures that increase the number of layers without loss in the generalization of the network and improve the accuracy results, such as residual blocks [19], and dense blocks [129]. This is another research challenge that can be considered in CELM architectures, increasing the number of layers to increase the accuracy without losing the network's generalization capacity. These deep convolutional approaches should inspire CNNs' architectures for CELM.

There is also a research field that aims to compact deep learning models, which accelerate the learning process. Compressing CNNs can provide lightweight architectures for applications which demand high computational cost. Two well-known techniques used for CNN compression are pruning and weight quantization. The pruning process handles removing a subset of parameters (filters, layers, or weights) evaluated as less critical for the task. None of the works reported in this systematic review reported the use of pruning or weight quantization. Approaches for pruning or weight quantization (or a combination of both) could improve the learning process of CELMs, removing irrelevant information in the neural network and optimizing the support for real-time applications.

In this systematic review, we did not report any work on object detection problems. Deep learning research field architectures for object detection such as R-CNN, Mask R-CNN, and YOLO could inspire new CELM works. Such architectures have high computational costs. When the object detection deep learning models are processed into the CPU, there is a loss in computational performance. Developing new architectures for object detection using ELM concepts could help such applications where computational resources are limited.

Another common computer vision problem recurring in the literature and little addressed in this systematic review is semantic segmentation. The difficulty may be linked to image reconstruction and decoding operations through deconvolutions usually done through the backpropagation algorithm. This is another open challenge in CELM, where ELM networks could replace the backpropagation in the calculation of updating the weights of both convolutional and deconvolutional layers for the reconstruction of the segmented images.

Despite presenting promising and interesting results in RGB-D classification and remote sensing tasks, there is a lack with CELM networks. There are not yet works that prove the strength of CELM in very large datasets for even more complex tasks. Therefore, there is a need for performance evaluation (accuracy and computation) of CELM models on the large current state of the art, such as ImageNet, COCO dataset, and Pascal-VOC. These last three cited databases are current references in deep learning for image classification, object detection, and semantic segmentation, in addition to other problems such as detection of human poses and panoptic segmentation, and so on. The performing of new experiments on the state of the art's datasets in deep learning can strengthen all aspects of CELM advantages covered in this systematic review.

8. Conclusion

We presented a systematic review on convolutional extreme learning machines in multimedia analysis based on four research questions. We found four different types of CELM architectures, such as: (i) CNN with predefined kernels for feature extraction and ELM for fast training; (ii) pre-trained CNN in other application domain for feature extraction and ELM for fast training; (iii) pre-trained CNN in same application domain for feature extraction and ELM for fast training; and (iv) fast training of CNNs using ELM concepts. We reported 19 different scenarios evaluated with different datasets, such as object recognition (NORB dataset), remote sensing classification (Pavia dataset), different medicine applications, handwritten digit recognition (MNIST dataset), face recognition (YALE dataset), and RGB-D image recognition (Washington RGB-D Object dataset).

Analyzing the results found in the primary studies, we can state that CELM models provide good accuracy and good computational performance. We highlight the excellent feature representation achieved by CELM, which can explain its good accuracy results. In general, the CELM architectures present fast and good convergence by changing the conventional fully connected layers to the ELM network. With this change, the convergence speed increases and avoids fine adjustments by the backpropagation algorithm's iterations. Finally, there is a decrease in the total processing time required for the learning process in CELM architectures, making it suitable to solve image analysis problems on a large scale, such as real-time or big data applications.

Acknowledgments

This work was supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *nature* **2015**, *521*, 436–444.
2. Rawat, W.; Wang, Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation* **2017**, *29*, 2352–2449.
3. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
4. Guo, Y.; Liu, Y.; Georgiou, T.; Lew, M.S. A review of semantic segmentation using deep neural networks. *International journal of multimedia information retrieval* **2018**, *7*, 87–93.
5. Shen, D.; Wu, G.; Suk, H.I. Deep learning in medical image analysis. *Annual review of biomedical engineering* **2017**, *19*, 221–248.

6. Huang, G.; Bai, Z.; Kasun, L.L.C.; Vong, C.M. Local Receptive Fields Based Extreme Learning Machine. *IEEE Computational Intelligence Magazine* **2015**, *10*, 18–29.
7. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: theory and applications. *Neurocomputing* **2006**, *70*, 489–501.
8. Cao, J.; Lin, Z. Extreme learning machines on high dimensional and large data applications: a survey. *Mathematical Problems in Engineering* **2015**, *2015*.
9. Akusok, A.; Björk, K.M.; Miche, Y.; Lendasse, A. High-performance extreme learning machines: a complete toolbox for big data applications. *IEEE Access* **2015**, *3*, 1011–1025.
10. dos Santos, M.M.; da Silva Filho, A.G.; dos Santos, W.P. Deep convolutional extreme learning machines: filters combination and error model validation. *Neurocomputing* **2019**, *329*, 359–369.
11. Huang, G.B.; Wang, D.H.; Lan, Y. Extreme learning machines: a survey. *International journal of machine learning and cybernetics* **2011**, *2*, 107–122.
12. Huang, G.; Huang, G.B.; Song, S.; You, K. Trends in extreme learning machines: A review. *Neural Networks* **2015**, *61*, 32–48.
13. Salaken, S.M.; Khosravi, A.; Nguyen, T.; Nahavandi, S. Extreme learning machine based transfer learning algorithms: A survey. *Neurocomputing* **2017**, *267*, 516–524.
14. Zhang, J.; Li, Y.; Xiao, W.; Zhang, Z. Non-iterative and Fast Deep Learning: Multilayer Extreme Learning Machines. *Journal of the Franklin Institute* **2020**, *357*, 8925–8955.
15. Endo, P.T.; Rodrigues, M.; Gonçalves, G.E.; Kelner, J.; Sadok, D.H.; Curescu, C. High availability in clouds: systematic review and research challenges. *Journal of Cloud Computing* **2016**, *5*, 16.
16. Coutinho, E.F.; de Carvalho Sousa, F.R.; Rego, P.A.L.; Gomes, D.G.; de Souza, J.N. Elasticity in cloud computing: a survey. *annals of telecommunications-annales des télécommunications* **2015**, *70*, 289–309.
17. Kitchenham, B. Procedures for performing systematic reviews. *Keele, UK, Keele University* **2004**, *33*, 1–26.
18. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.
19. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
20. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. JMLR.org, 2015, ICML'15.
21. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **2014**, *15*, 1929–1958.
22. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; others. Recent advances in convolutional neural networks. *Pattern Recognition* **2018**, *77*, 354–377.
23. Xu, X.; Li, G.; Xie, G.; Ren, J.; Xie, X. Weakly supervised deep semantic segmentation using CNN and ELM with semantic candidate regions. *Complexity* **2019**, *2019*.
24. Yoo, Y.; Oh, S.Y. Fast training of convolutional neural network classifiers through extreme learning machines. 2016 International Joint Conference on Neural Networks (IJCNN). IEEE, 2016, pp. 1702–1708.
25. Bai, Z.; Kasun, L.; Huang, G.B. Generic Object Recognition with Local Receptive Fields Based Extreme Learning Machine. *Procedia Computer Science* **2015**, *53*, 391 – 399. INNS Conference on Big Data 2015 Program San Francisco, CA, USA 8-10 August 2015.
26. He, B.; Song, Y.; Zhu, Y.; Sha, Q.; Shen, Y.; Yan, T.; Nian, R.; Lendasse, A. Local receptive fields based extreme learning machine with hybrid filter kernels for image classification. *Multidimensional systems and signal processing* **2019**, *30*, 1149–1169.
27. Wu, C.; Li, Y.; Zhao, Z.; Liu, B. Extreme learning machine with autoencoding receptive fields for image classification. *Neural Computing and Applications* **2020**, *32*, 8157–8173.
28. Wu, C.; Li, Y.; Zhao, Z.; Liu, B. Extreme learning machine with multi-structure and auto encoding receptive fields for image classification. *Multidimensional Systems and Signal Processing* **2020**, pp. 1–22.
29. Song, G.; Dai, Q.; Han, X.; Guo, L. Two novel ELM-based stacking deep models focused on image recognition. *Applied Intelligence* **2020**, pp. 1–22.
30. Chang, P.; Zhang, J.; Wang, J.; Fei, R. ELMAENet: A Simple, Effective and Fast Deep Architecture for Image Classification. *Neural Processing Letters* **2020**, *51*, 129–146.

31. Alshalali, T.; Josyula, D. Fine-Tuning of Pre-Trained Deep Learning Models with Extreme Learning Machine. 2018 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE, 2018, pp. 469–473.
32. Han, J.S.; Cho, G.B.; Kwak, K.C. A Design of Convolutional Neural Network Using ReLU-Based ELM Classifier and Its Application. Proceedings of the 9th International Conference on Machine Learning and Computing, 2017, pp. 179–183.
33. Hao, P.; Zhai, J.H.; Zhang, S.F. A simple and effective method for image classification. 2017 International Conference on Machine Learning and Cybernetics (ICMLC). IEEE, 2017, Vol. 1, pp. 230–235.
34. Cui, D.; Zhang, G.; Han, W.; Lekamalage Chamara Kasun, L.; Hu, K.; Huang, G.B. Compact feature representation for image classification using elms. Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 1015–1022.
35. Zhu, X.; Li, Z.; Zhang, X.Y.; Li, P.; Xue, Z.; Wang, L. Deep convolutional representations and kernel extreme learning machines for image classification. *Multimedia Tools and Applications* **2019**, *78*, 29271–29290.
36. Zhang, L.; He, Z.; Liu, Y. Deep object recognition across domains based on adaptive extreme learning machine. *Neurocomputing* **2017**, *239*, 194–203.
37. Zhang, L.; Zhang, D.; Tian, F. SVM and ELM: Who Wins? Object recognition with deep convolutional features from ImageNet. In *Proceedings of ELM-2015 Volume 1*; Springer, 2016; pp. 249–263.
38. Liu, H.; Li, F.; Xu, X.; Sun, F. Active object recognition using hierarchical local-receptive-field-based extreme learning machine. *Memetic Computing* **2018**, *10*, 233–241.
39. He, X.; Liu, H.; Huang, W. Room categorization using local receptive fields-based extreme learning machine. 2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM). IEEE, 2017, pp. 620–625.
40. LeCun, Y.; Huang, F.J.; Bottou, L. Learning methods for generic object recognition with invariance to pose and lighting. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. IEEE, 2004, Vol. 2, pp. II–104.
41. Krizhevsky, A.; Hinton, G.; others. Learning multiple layers of features from tiny images **2009**.
42. Nene, S.A.; Nayar, S.K.; Murase, H.; others. Columbia object image library (coil-100) **1996**.
43. Fei-Fei, L.; Fergus, R.; Perona, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. 2004 conference on computer vision and pattern recognition workshop. IEEE, 2004, pp. 178–178.
44. Leibe, B.; Schiele, B. Analyzing appearance and contour based methods for object categorization. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. IEEE, 2003, Vol. 2, pp. II–409.
45. Ma, L.; Liu, Y.; Zhang, X.; Ye, Y.; Yin, G.; Johnson, B.A. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS journal of photogrammetry and remote sensing* **2019**, *152*, 166–177.
46. Huang, F.; Lu, J.; Tao, J.; Li, L.; Tan, X.; Liu, P. Research on Optimization Methods of ELM Classification Algorithm for Hyperspectral Remote Sensing Images. *IEEE Access* **2019**, *7*, 108070–108089.
47. Shen, Y.; Xiao, L.; Chen, J.; Pan, D. A Spectral-Spatial Domain-Specific Convolutional Deep Extreme Learning Machine for Supervised Hyperspectral Image Classification. *IEEE Access* **2019**, *7*, 132240–132252.
48. Shi, J.; Ku, J. Spectral-spatial classification of hyperspectral image using distributed extreme learning machine with MapReduce. 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(. IEEE, 2017, pp. 714–720.
49. Cao, F.; Yang, Z.; Ren, J.; Ling, B.W.K. Convolutional neural network extreme learning machine for effective classification of hyperspectral images. *Journal of Applied Remote Sensing* **2018**, *12*, 035003.
50. Li, J.; Zhao, X.; Li, Y.; Du, Q.; Xi, B.; Hu, J. Classification of hyperspectral imagery using a new fully convolutional neural network. *IEEE Geoscience and Remote Sensing Letters* **2018**, *15*, 292–296.
51. Lv, Q.; Niu, X.; Dou, Y.; Xu, J.; Lei, Y. Classification of hyperspectral remote sensing image using hierarchical local-receptive-field-based extreme learning machine. *IEEE Geoscience and Remote Sensing Letters* **2016**, *13*, 434–438.
52. Lv, Q.; Niu, X.; Dou, Y.; Wang, Y.; Xu, J.; Zhou, J. Hyperspectral image classification via kernel extreme learning machine using local receptive fields. 2016 IEEE International Conference on Image Processing (ICIP). IEEE, 2016, pp. 256–260.

53. Lv, Q.; Niu, X.; Dou, Y.; Xu, J.; Xia, F. Leveraging local receptive fields based random weights networks for hyperspectral image classification. *Journal of Intelligent & Fuzzy Systems* **2016**, *31*, 1017–1028.
54. Shen, Y.; Chen, J.; Xiao, L. Supervised classification of hyperspectral images using local-receptive-fields-based kernel extreme learning machine. 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017, pp. 3120–3124.
55. Gu, Y.; Xu, Y.; Liu, J. SAR ATR by Decision Fusion of Multiple Random Convolution Features. 2019 22th International Conference on Information Fusion (FUSION). IEEE, 2019, pp. 1–8.
56. Wang, P.; Zhang, X.; Hao, Y. A Method Combining CNN and ELM for Feature Extraction and Classification of SAR Image. *Journal of Sensors* **2019**, *2019*.
57. Ye, L.; Wang, L.; Sun, Y.; Zhu, R.; Wei, Y. Aerial scene classification via an ensemble extreme learning machine classifier based on discriminative hybrid convolutional neural networks features. *International Journal of Remote Sensing* **2019**, *40*, 2759–2783.
58. Romay, D.M.G. Hyperspectral remote sensing scenes, 2020.
59. Keydel, E.R.; Lee, S.W.; Moore, J.T. MSTAR extended operating conditions: A tutorial. Algorithms for Synthetic Aperture Radar Imagery III. International Society for Optics and Photonics, 1996, Vol. 2757, pp. 228–242.
60. Coman, C.; others. A deep learning sar target classification experiment on mstar dataset. 2018 19th International Radar Symposium (IRS). IEEE, 2018, pp. 1–6.
61. Özyurt, F.; Sert, E.; Avcı, D. An expert system for brain tumor detection: Fuzzy C-means with super resolution and convolutional neural network with extreme learning machine. *Medical hypotheses* **2020**, *134*, 109433.
62. Pashaei, A.; Sajedi, H.; Jazayeri, N. Brain tumor classification via convolutional neural network and extreme learning machines. 2018 8th International conference on computer and knowledge engineering (ICCKE). IEEE, 2018, pp. 314–319.
63. Ari, A.; Hanbay, D. Deep learning based brain tumor classification and detection system. *Turkish Journal of Electrical Engineering & Computer Sciences* **2018**, *26*, 2275–2286.
64. Yu, J.s.; Chen, J.; Xiang, Z.; Zou, Y.X. A hybrid convolutional neural networks with extreme learning machine for WCE image classification. 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2015, pp. 1822–1827.
65. Doğantekin, A.; Özyurt, F.; Avcı, E.; Koç, M. A novel approach for liver image classification: PH-C-ELM. *Measurement* **2019**, *137*, 332–338.
66. Özyurt, F. A fused CNN model for WBC detection with MRMR feature selection and extreme learning machine. *Soft Computing* **2020**, *24*, 8163–8172.
67. Fang, J.; Xu, X.; Liu, H.; Sun, F. Local receptive field based extreme learning machine with three channels for histopathological image classification. *International Journal of Machine Learning and Cybernetics* **2019**, *10*, 1437–1447.
68. Lu, S.; Xia, K.; Wang, S.H. Diagnosis of cerebral microbleed via VGG and extreme learning machine trained by Gaussian map bat algorithm. *Journal of Ambient Intelligence and Humanized Computing* **2020**, pp. 1–12.
69. Ghoneim, A.; Muhammad, G.; Hossain, M.S. Cervical cancer classification using convolutional neural networks and extreme learning machines. *Future Generation Computer Systems* **2020**, *102*, 643–649.
70. Monkam, P.; Qi, S.; Xu, M.; Li, H.; Han, F.; Teng, Y.; Qian, W. Ensemble learning of multiple-view 3D-CNNs model for micro-nodules identification in CT images. *IEEE Access* **2018**, *7*, 5564–5576.
71. Fang, L.; Wang, C.; Li, S.; Yan, J.; Chen, X.; Rabbani, H. Automatic classification of retinal three-dimensional optical coherence tomography images using principal component analysis network with composite kernels. *Journal of Biomedical Optics* **2017**, *22*, 116011.
72. Li, S.; Jiang, H.; Pang, W. Joint multiple fully connected convolutional neural network with extreme learning machine for hepatocellular carcinoma nuclei grading. *Computers in Biology and Medicine* **2017**, *84*, 156–167.
73. Baldominos, A.; Saez, Y.; Isasi, P. A survey of handwritten character recognition with mnist and emnist. *Applied Sciences* **2019**, *9*, 3169.
74. Khellal, A.; Ma, H.; Fei, Q. Convolutional Neural Network Features Comparison Between Back-Propagation and Extreme Learning Machine. 2018 37th Chinese Control Conference (CCC). IEEE, 2018, pp. 9629–9634.
75. Kannojia, S.P.; Jaiswal, G. Ensemble of hybrid CNN-ELM model for image classification. 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN). IEEE, 2018, pp. 538–541.

76. Ding, S.; Guo, L.; Hou, Y. Extreme learning machine with kernel model based on deep learning. *Neural Computing and Applications* **2017**, *28*, 1975–1984.
77. Pang, S.; Yang, X. Deep convolutional extreme learning machine and its application in handwritten digit classification. *Computational intelligence and neuroscience* **2016**, *2016*.
78. LeCun, Y.; Cortes, C.; Burges, C. THE MNIST DATABASE: of handwritten digits. *http://yann.lecun.com/exdb/mnist/. Accessed in 25 Ago. 2020.* **1998**.
79. Hull, J.J. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence* **1994**, *16*, 550–554.
80. Hu, G.; Yang, Y.; Yi, D.; Kittler, J.; Christmas, W.; Li, S.Z.; Hospedales, T. When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition. Proceedings of the IEEE international conference on computer vision workshops, 2015, pp. 142–150.
81. Yu, D.; Wu, X.J. 2DPCANet: a deep leaning network for face recognition. *Multimedia Tools and Applications* **2018**, *77*, 12919–12934.
82. Ripon, K.S.N.; Ali, L.E.; Siddique, N.; Ma, J. Convolutional Neural Network based Eye Recognition from Distantly Acquired Face Images for Human Identification. 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019, pp. 1–8.
83. Wang, K.; Liu, M.; Hao, X.; Xing, X. Decision-Level Fusion Method Based on Deep Learning. Chinese Conference on Biometric Recognition. Springer, 2017, pp. 673–682.
84. Gürpınar, F.; Kaya, H.; Salah, A.A. Combining deep facial and ambient features for first impression estimation. European conference on computer vision. Springer, 2016, pp. 372–385.
85. Yale. The normalized yale face database. *https://vismod.media.mit.edu/vismod/classes/mas622-00/datasets/. Accessed in 25 Ago. 2020.* **1998**.
86. Hoyer, P.O. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research* **2004**, *5*, 1457–1469.
87. Cai, Z.; Han, J.; Liu, L.; Shao, L. RGB-D datasets using microsoft kinect or similar sensors: a survey. *Multimedia Tools and Applications* **2017**, *76*, 4313–4355.
88. Wang, P.; Li, W.; Ogunbona, P.; Wan, J.; Escalera, S. RGB-D-based human motion recognition with deep learning: A survey. *Computer Vision and Image Understanding* **2018**, *171*, 118–139.
89. Shao, L.; Han, J.; Kohli, P.; Zhang, Z. *Computer vision and machine learning with RGB-D sensors*; Vol. 20, Springer, 2014.
90. Boubou, S.; Narikiyo, T.; Kawanishi, M. Object recognition from 3d depth data with extreme learning machine and local receptive field. 2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM). IEEE, 2017, pp. 394–399.
91. Liu, H.; Li, F.; Xu, X.; Sun, F. Multi-modal local receptive field extreme learning machine for object recognition. *Neurocomputing* **2018**, *277*, 4–11.
92. Yin, Y.; Li, H. Multi-view CSPMPR-ELM feature learning and classifying for RGB-D object recognition. *Cluster Computing* **2019**, *22*, 8181–8191.
93. Yin, Y.; Li, H. RGB-D object recognition based on the joint deep random kernel convolution and ELM. *Journal of Ambient Intelligence and Humanized Computing* **2018**, pp. 1–10.
94. Zaki, H.F.; Shafait, F.; Mian, A. Viewpoint invariant semantic object and scene categorization with RGB-D sensors. *Autonomous Robots* **2019**, *43*, 1005–1022.
95. Yin, Y.; Li, H.; Wen, X. Multi-model convolutional extreme learning machine with kernel for RGB-D object recognition. LIDAR Imaging Detection and Target Recognition 2017. International Society for Optics and Photonics, 2017, Vol. 10605, p. 106051Z.
96. Yang, Z.X.; Tang, L.; Zhang, K.; Wong, P.K. Multi-view cnn feature aggregation with elm auto-encoder for 3d shape recognition. *Cognitive Computation* **2018**, *10*, 908–921.
97. Ijjina, E.P.; Chalavadi, K.M. Human action recognition in RGB-D videos using motion sequence information and deep learning. *Pattern Recognition* **2017**, *72*, 504–516.
98. Lai, K.; Bo, L.; Ren, X.; Fox, D. A large-scale hierarchical multi-view rgb-d object dataset. 2011 IEEE international conference on robotics and automation. IEEE, 2011, pp. 1817–1824.
99. Martinel, N.; Piciarelli, C.; Foresti, G.L.; Micheloni, C. Mobile food recognition with an extreme deep tree. Proceedings of the 10th International Conference on Distributed Smart Camera, 2016, pp. 56–61.
100. Li, Z.; Zhu, X.; Wang, L.; Guo, P. Image classification using convolutional neural networks and kernel extreme learning machines. 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018, pp. 3009–3013.

101. Horii, K.; Maeda, K.; Ogawa, T.; Haseyama, M. A Human-Centered Neural Network Model with Discriminative Locality Preserving Canonical Correlation Analysis for Image Classification. 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018, pp. 2366–2370.
102. Pashaei, A.; Ghatee, M.; Sajedi, H. Convolution neural network joint with mixture of extreme learning machines for feature extraction and classification of accident images. *Journal of Real-Time Image Processing* **2020**, *17*, 1051–1066.
103. Zeng, Y.; Xu, X.; Fang, Y.; Zhao, K. Traffic sign recognition using deep convolutional networks and extreme learning machine. International Conference on Intelligent Science and Big Data Engineering. Springer, 2015, pp. 272–280.
104. Zhou, Y.; Liu, Q.; Zhao, Y.; Li, W. Aluminum Foil Packaging Sealing Testing Method Based on Gabor Wavelet and ELM Neural Network. Proceedings of the 2nd International Conference on Advances in Image Processing, 2018, pp. 59–63.
105. Liu, H.; Fang, J.; Xu, X.; Sun, F. Surface material recognition using active multi-modal extreme learning machine. *Cognitive Computation* **2018**, *10*, 937–950.
106. Xu, X.; Fang, J.; Li, Q.; Xie, G.; Xie, J.; Ren, M. Multi-scale local receptive field based online sequential extreme learning machine for material classification. International Conference on Cognitive Systems and Signal Processing. Springer, 2018, pp. 37–53.
107. Zhang, Y.; Zhang, L.; Li, P. A novel biologically inspired ELM-based network for image recognition. *Neurocomputing* **2016**, *174*, 286–298.
108. Imran, J.; Raman, B. Deep motion templates and extreme learning machine for sign language recognition. *The Visual Computer* **2020**, *36*, 1233–1246.
109. Xie, X.; Guo, W.; Jiang, T. Body Gestures Recognition Based on CNN-ELM Using Wi-Fi Long Preamble. International Conference in Communications, Signal Processing, and Systems. Springer, 2018, pp. 877–889.
110. Sun, R.; Wang, X.; Yan, X. Robust visual tracking based on convolutional neural network with extreme learning machine. *Multimedia Tools and Applications* **2019**, *78*, 7543–7562.
111. Huang, J.; Yu, Z.L.; Cai, Z.; Gu, Z.; Cai, Z.; Gao, W.; Yu, S.; Du, Q. Extreme learning machine with multi-scale local receptive fields for texture classification. *Multidimensional Systems and Signal Processing* **2017**, *28*, 995–1011.
112. Kölsch, A.; Afzal, M.Z.; Ebbecke, M.; Liwicki, M. Real-time document image classification using deep CNN and extreme learning machines. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2017, Vol. 1, pp. 1318–1323.
113. Li, D.; Qiu, X.; Zhu, Z.; Liu, Y. Criminal Investigation Image Classification Based on Spatial CNN Features and ELM. 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC). IEEE, 2018, Vol. 2, pp. 294–298.
114. Li, F.; Liu, H.; Xu, X.; Sun, F. Haptic recognition using hierarchical extreme learning machine with local-receptive-field. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 541–547.
115. Sharma, J.; Granmo, O.C.; Goodwin, M. Deep CNN-ELM Hybrid Models for Fire Detection in Images. International Conference on Artificial Neural Networks. Springer, 2018, pp. 245–259.
116. Li, R.; Lu, W.; Liang, H.; Mao, Y.; Wang, X. Multiple features with extreme learning machines for clothing image recognition. *IEEE Access* **2018**, *6*, 36283–36294.
117. Yang, Y.; Li, D.; Duan, Z. Chinese vehicle license plate recognition using kernel-based extreme learning machine with deep convolutional features. *IET Intelligent Transport Systems* **2017**, *12*, 213–219.
118. Kittler, J.; Hatef, M.; Duin, R.P.; Matas, J. On combining classifiers. *IEEE transactions on pattern analysis and machine intelligence* **1998**, *20*, 226–239.
119. Chan, T.H.; Jia, K.; Gao, S.; Lu, J.; Zeng, Z.; Ma, Y. PCANet: A simple deep learning baseline for image classification? *IEEE transactions on image processing* **2015**, *24*, 5017–5032.
120. Afridi, M.J.; Ross, A.; Shapiro, E.M. On automated source selection for transfer learning in convolutional neural networks. *Pattern recognition* **2018**, *73*, 65–75.
121. Han, D.; Liu, Q.; Fan, W. A new image classification method using CNN transfer learning and web data augmentation. *Expert Systems with Applications* **2018**, *95*, 43–56.
122. Horii, K.; Maeda, K.; Ogawa, T.; Haseyama, M. Human-centered image classification via a neural network considering visual and biological features. *Multimedia Tools and Applications* **2020**, *79*, 4395–4415.
123. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 2012, pp. 1097–1105.

doi:10.20944/preprints202104.0753.v1

124. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. Proceedings of the 22nd ACM international conference on Multimedia, 2014, pp. 675–678.
125. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
126. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.
127. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360* **2016**.
128. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* **2017**.
129. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
130. Park, Y.; Yang, H.S. Convolutional neural network based on an extreme learning machine for image classification. *Neurocomputing* **2019**, *339*, 66–76.
131. Khellal, A.; Ma, H.; Fei, Q. Convolutional neural network based on extreme learning machine for maritime ships recognition in infrared images. *Sensors* **2018**, *18*, 1490.
132. Kim, J.; Kim, J.; Jang, G.J.; Lee, M. Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection. *Neural Networks* **2017**, *87*, 109–121.