

# Neural Network Design using a Virtual Reality Platform

Francesco Carlo Morabito

*Received: 1 January 1970 Accepted: 1 January 1970 Published: 1 January 1970*

---

## Abstract

The evolution of Deep Learning (DL), a subset of machine learning, has made their use very effective in many artificial intelligence (AI) fields. In parallel Virtual Reality is going wide in many applications thanks to the proliferation of cameras in mobile devices and improved processing efficiency. Data visualization in deep learning is a fundamental element for which it can benefit from the advantages offered by the visualization of the VR for the development of the models. In addition, the researchers can widely use the editing of images and videos in the machine learning process to design a convolutional network suitable for image recognition. In this study, we want to demonstrate the usefulness of this approach in collecting data within virtual reality to train and optimize a convolutional neural network used to recognize human activities (HAR).

---

*Index terms*— machine learning, deep learning, neural nets, visualization, virtual reality.

## 1 I. Introduction

Machine Learning (ML) indicates a research area within Artificial Intelligence needs to create systems capable of autonomous learning without being specifically programmed to carry out this task [1]. This step introduces a new programming vision, in which the goal is no longer to feed input to receive an output from the machine but to allow it to make decisions autonomously.

Deep Learning (DL) is a machine learning technique used to build Artificial Intelligence (AI) systems [2]. It is a technology that creates learning models on several levels. Deep learning systems allow a representation of information at various levels in a hierarchical way and build models that exploit large amounts of visual media data. The learning of data takes place through the use of statistical calculation algorithms. The inspiring principle was to reproduce a system of reasoning that is biologically inspired by the human brain, must be able to behave by simulating the functioning of neurons. The human neuron represents the computational engine that is the basis of deep learning that takes place in Artificial Neural Networks (ANN). A neural network reproduces all those processes in the brain during the learning phase and subsequent recognition phase. We design the artificial neural networks (ANN) to perform complex analysis of large amounts of data by passing it through multiple layers of neurons. The peculiar characteristic of neural networks is learning and adapting to their environment. Another property is to build connectivity models based on the approximation of the error. These elements make identifying similar patterns in the test data [3].

A neural network requires a training phase and an inference phase to operate correctly. During the training phase, the number of neurons and the levels that will comprise the neural network are defined and exposed to the labeled training data. Finally, the network will learn which data is correct and discard the wrong one. Once the network has been trained, it is possible to proceed to the inference phase in which the network itself is called to evaluate new images proposed to it. These steps are part of the neural network design process, which includes: 1) Determination of the function that the network must be able to perform: a) Classification, this function involves sorting images into various classes and grouping them according to common properties, allowing you to determine whether a specific type of object is present in an image or not; b) Detection and localization, this function allows you to identify the position and size of an object by identifying the characteristics of the image; c) Segmentation, used to identify which pixels of an image belong to the corresponding objects, allows determining the boundaries and areas of the objects in the images.

46 2) Choice of framework: we must make this choice based on the applications to be created, the libraries  
47 available, or the developers' competence.

48 In addition, there are numerous accessible frameworks such as, e.g., TensorFlow of Google, Caffe2 of Facebook  
49 e OpenVino of Intel, and

50 PyTorch: an open-source solution that is now part of Facebook.

51 3) Loading training data: this phase and the training are crucial for the proper functioning of the neural  
52 networks. While being favored by the ability to generate features as they learn automatically, deep networks  
53 still require large amounts of training data to develop a properly functioning model. The amount of data needed  
54 depends on the complexity of the domain we are trying to approximate. The choice of the size of the dataset is  
55 closely related to the selection of the number of neurons in the neural network. The total complexity is generally  
56 not known before starting the training. For this reason, the Year 2022

57 neural network training process is iterative. At the end of the training, the network's performance is analyzed:  
58 the results of this analysis provide information on whether the available data is sufficient. The data for training  
59 can be found already labeled on public archives or purchased. Alternatively, it is necessary to generate a faked  
60 data set and label them. Each type of application to be developed requires its own set of data, complete  
61 with all the characteristics used for a correct evaluation. There is a wide variety of deep neural networks  
62 (DNN). Convolutional neural networks (CNN) or Deep convolutional neural networks (DCNN) are the types most  
63 commonly used to identify patterns in images and video. The name "convolutional neural network" indicates  
64 that it employs a mathematical operation called convolution in place of general matrix multiplication in at least  
65 one of their layers [4]. They have evolved from traditional artificial neural networks, applying the connectivity  
66 pattern between neurons that resembles the organization of the animal visual cortex. Individual cortical neurons  
67 respond to stimuli only in a restricted visual field region known as the receptive field. The receptive fields of  
68 different neurons partially overlap such that they cover the entire visible area. CNN's are regularized versions of  
69 multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks; each neuron in one layer is  
70 connected to all neurons in the next layer. Deep convolutional neural networks mainly focus on object detection,  
71 image classification, and recommendation systems and are sometimes used for natural language processing. In  
72 the absence of CNN to identify the objects in the images, it was necessary to resort to manual feature extraction  
73 methods, making mistakes and taking a considerable amount of time. CNN's are characterized by their ability  
74 to optimize filters (or Kernel) through machine learning, differentiating themselves from traditional algorithms  
75 in which filters must be manually sized.

76 A feature common to all architectures of deep neural networks, including CNN's, is their high complexity due  
77 to many internal parameters. For this reason, numerous techniques have been developed to facilitate the design  
78 and understanding of the internal workings of a network [5]. The lack of interpretability and transparency of  
79 neural networks and the absence of explanations on the decisions taken can compromise the confidence in their  
80 applicability. Moreover, they are endorsed by the large datasets required to train most deep learning models.  
81 Therefore, the possibility of understanding the model's functioning allows one to overcome these problems.  
82 Finally, we can understand how deep learning models make decisions and what representations they have learned  
83 through deep learning visualization. In this way, we regain confidence in the model [6]. They used a method based  
84 on analyzing a new type of training error curve on three levels of detail. Usually, the comparison method between  
85 their training error curves is used to evaluate the learning processes resulting from different hyperparameter  
86 settings. With this approach, the authors associate three levels of detail to the three stages of Shneiderman's  
87 Visual Information Seeking Mantra. In the first stage, the traditional training error curve is the focus of attention.  
88 In the zoom and filter stage, we can explore the variation of the training error along each of the descent directions  
89 considered during the gradient descent-based DNN training. The third stage allows for an in-depth exploration  
90 of methods for adaptive control of the step width along a given descent direction.

## 91 2 II. Data Visualization

92 Other techniques that focus visualization on the learning process rather than the magnitude of the error are Deep  
93 Tracker [9] and Tensor view [10].

94 Deep Tracker is a visual analysis system designed to analyze CNN training processes' dynamics and identify  
95 unusual patterns within the numerous training logs. The authors associated a series of small hierarchical multiples  
96 with a hierarchical structure index mechanism to allow you to explore the entire training process from different  
97 levels of detail. They also suggest a new cube-style visualization to reveal the complex correlations across multiple  
98 types of heterogeneous training data.

99 Tensor View is a tool that uses Paraview and Matplotlib to study neural networks built on the TensorFlow  
100 framework. It allows visualizing the evolution of the neural network parameters, such as the weights of the  
101 filters and gradients, providing direct observation of how the neurons update during training. This tool uses  
102 visualization to inform researchers about their hyperparameters; to use quantitative information to eliminate  
103 unnecessary neurons, accelerating the training processes.

104 These two tools are very complex, so expert researchers can only use them. Other visualization techniques aim  
105 to provide the right amount of information on the evolution of the training process, facilitating the understanding  
106 of even less experienced researchers.

107 Michelle Peters and al. [11] developed, for nonexperts, a technique that intuitively visualizes the training

---

108 process of a neural network to facilitate the understanding of the decisions of a neural network. They use the  
109 dimension reduction method Uniform Manifold Approximation and Projection (UMAP) [12] to visualize neuronal  
110 activity. They generate a visualization with three plots: 1) -a 2D plot of the test data with the input images, 2)  
111 -a 2D plot of the training data, 3) -a line plot of the accuracy of both the testing and training data. They also  
112 create a video, generating a frame for every epoch, showing how the neural network improves during the training  
113 phase. This approach demonstrates how an ongoing training video offers more information than a static view  
114 at the end of the process and which features are helpful for non-experts to view, highlighting that the insights  
115 drawn from video analysis can be beneficial for network design.

116 Various visualization techniques have been incorporated into interactive tools to aid in the DNN design process.

117 Bock et al. [13] designed a tool for 3D representation of deep learning algorithms for experts and non-experts  
118 with an interactive user interface that allows visualiz Pezzotti and al. [14] developed a progressive visual analysis  
119 technique called Deepeyes. Multiple linked views are associated with analyzing a network during training and  
120 showing how it changes over time. In addition, the system supports researchers in identifying problems, such as  
121 unnecessary filters or layers and information not captured by the network.

122 Kahng et al. [15] developed ActiVis, an interactive visualization system for large-scale deep neural networks,  
123 to help users understand how a model derives its predictions. Users can better diagnose discrepancies with  
124 their neural network by integrating multiple coordinated views: such as an overview of the model architecture  
125 computation graph, of how the deep learning neurons, which together make up the network, are activated by user-  
126 specified instances or subsets of instances. Finally, it provides a graphic representation of the model architecture;  
127 hence the user can perform a localized inspection of activations on each model level.

128 A classification of CNN viewing techniques distinguishes them into two categories: feature display and  
129 attribution [16]. With the feature visualization technique, we study how the network operates to generate  
130 an example; attribution studies which part of the example is responsible for attributing the network in a certain  
131 way.

132 One of the methods for feature visualization is activation maximization [17] which can be applied at different  
133 levels of a CNN. This technique highlights that a CNN tends to build its understanding of an image hierarchically  
134 on many levels. For example, Nguyen and al. have developed a system based on multifaceted feature visualization  
135 (MFV) [18], uncovering the different types of features learned by each neuron in deep neural networks. Previous  
136 activation maximization techniques constructed images without regard for the multiple facets of a neuron; instead,  
137 this method explicitly shows each neuron's various aspects by producing a synthetic visualization of each of the  
138 types of images that activate a neuron. The visualization of the faceted features thus provides a more accurate  
139 and more complete description of the role of each neuron.

### 140 3 ( ) D Year 2022

141 To attribution methods, Chattopadhyay and al. [19] have developed a methodology that starts from the  
142 visualization of the neural network's architecture as a Structural Causal Model (SCM) and calculates each  
143 feature's causal effect on the output. Their approach differs from other related works to structural learning [20].  
144 The goal is to discern the causal structure in data provided; their goal is to identify the causal influence on the  
145 output of a learned function.

146 Sattarzadeh and al. [21] propose a model based on visualization maps from multiple levels using an attribution-  
147 based input sampling technique and aggregating them to achieve a satisfactory and complete explanation. The  
148 methodology adopted to interpret the layers of CNN results in a four-step method of description. In the first three  
149 stages, information extracted from multiple levels of CNN is represented in the accompanying display maps. These  
150 maps are then merged into a single module to form an explanation map in the last step. The proposed solution  
151 allows overcoming some problems encountered in models that use input sampling techniques. These methods  
152 have shown excellent fidelity in rationally inferring model predictions. However, they exhibit instability indices  
153 since their output depends on sampling randomness (RISE) or random initialization to optimize a perturbation  
154 mask. Finally, their algorithm called Semantic Input Sampling for Explanation (SISI) replaces the randomized  
155 RISE input sampling technique with an attribution-based sampling technique. Finally, it uses feature maps  
156 derived from multiple levels.

157 The Deep Replay open-source Python package was designed to allow you to visualize how a Deep Learning  
158 model in Keras is performing or learning at each iteration/epoch.

159 Virtual reality (VR), thanks to its peculiar characteristics, represents a valid alternative to the visualization of  
160 conventional data. VR technology establishes a new form of human-computer interaction, arousing in the user a  
161 new type of experience linked to the concept of "presence" [22]. Presence is the sensation of being physically and  
162 spatially placed in an environment. With virtual reality, you can realize this feeling of being in another world, the  
163 virtual one. VR can generate experience and support knowledge acquisition thanks to this new communication  
164 process. However, the sense of presence developed by VR depends significantly on the characteristics of the  
165 technology used. It is associated with the perceptual illusion that the user has to interact with the remote  
166 environment, such as if it were present. In the interaction with the VR environment, a perceptual illusion is  
167 created in the user. The stimulation of the senses produces cognitive and emotional models consistent with the  
168 experimenting environment. When in VR, a user will inevitably compare the appearance of virtual objects with  
169 real-world objects and judge the level of congruence. The other element that favors VR over other visualization

170 techniques is the level of "immersion" [23]. The greater immersion allows you to improve the design business,  
171 perform tasks more efficiently and with greater understanding. Immersion in virtual reality is the perception of  
172 being physically present in a non-physical world. Perception is created by surrounding the user of the VR system  
173 with images, sounds, or other stimuli that provide an evocative environment. Finally, the user can move freely  
174 and explore places and spaces through the movements of the head and eyes and interact with objects by grabbing  
175 and dragging them. In addition, movements from the physical world are transferred to the virtual world with  
176 great precision. This aspect is precisely exploited to simulate physical actions. Finally, workflows and activities  
177 can be transferred to the virtual model to verify the correct interpretation, improving the system's accuracy.

178 These characteristics make virtual reality a suitable environment for simulations and e-learning purposes as  
179 they can stimulate users' creativity and accelerate the learning process [24]. This technology is widely applied in  
180 various fields ranging from entertainment design to education and rehabilitation and psychological therapies [25].  
181 VR technology makes it possible to improve the rehabilitation-psychological program compared to traditional  
182 techniques by creating a more natural process, reproducing the characteristics of home living environments.  
183 Other elements that characterize this technique are represented by the possibility of dynamically adjusting the  
184 difficulty of the exercises concerning the skills acquired and stimulating the patient's multisensory skills. In the  
185 context of training, it has proved to be very effective compared to traditional techniques as they can keep the  
186 attention high and focus the topics better [26].

187 Immersion and interactivity are the elements that suggest the choice of VR for CNN as through them; the  
188 user can better focus and understand how it works.

189 The following sections illustrate the main characteristics of CNN and virtual reality technologies.

## 190 4 III. Convolutional Neural Network

191 From an architectural point of view, a convolutional neural network is a multi-layered feedforward neural network  
192 composed of an input layer, hidden layers, and an output layer. The hidden layers are typically convolutional,  
193 followed by activation and pooling layers. This sequential design allows convolutional neural networks to learn  
194 hierarchical features. Convolutional neural networks (CNN) process data through many layers of artificial  
195 neurons. This human brain (CNN) process is constituted by a set of different layers that act as extractors  
196 of the features of the input images and a fully connected terminal network that acts as a classifier. It has proved  
197 to be an effective Neural Network Design using a Virtual Reality Platform solution for image recognition [27].  
198 They are built to analyze images included within certain data sets and classify objects in images within them.  
199 CNN is a network composed of several convolutional layers [28]. Each processing layer comprises a convolutional  
200 filter, an activation function (ReLU), a pooling function, and a fully connected layer. At the end of each processing  
201 step, an input is generated for the next layer. In the convolutional operations, the trained filter set is convolved  
202 with input images to extract the specific feature to create the feature map, which becomes the input for the  
203 next filter. Finally, the design of a CNN network requires a training period followed by a test phase. During the  
204 training phase, the images are labeled and transferred to the subsequent layers to allow the structure to convert  
205 from the representation level of the original input to a higher level and more abstract representation to constitute  
206 the reference feature maps with which the network must compare the output feature maps. Once the training  
207 and test phase of the network has been completed, we will determine the survey's accuracy level.

208 Each layer comprises three levels: Convolution, ReLu, and Pooling.

209 ? Convolutional Level (CONV) is the main level of the network. Its objective is to identify patterns. They  
210 are multiple, designed to identify features present in the initial image. Each layer learns to extract specific parts  
211 of the photos placed at its entrance. Multiple layers in cascade combine the features of the previous layers with  
212 higher programmed extraction levels. ? Rectified Linear Unit (ReLU) Level is placed after the convolutional  
213 level and can cancel negative values obtained in the previous classes. ? Pool level allows identifying if the study  
214 characteristic is present in the previous story. The pooling layer obtains images with a particular resolution at  
215 the input and returns the same number of pictures with fewer pixels. The result of the convolution operations  
216 is the production of feature maps obtained with the help of filters that are matrices containing proper values for  
217 finding specific characteristics in the input images. At the end of the sequence of convolutional layers, there is  
218 then the fully connected level (FC) which aims to establish the identifying classes obtained in the previous levels  
219 according to a certain probability.

220 Each category represents a possible answer that the system will most likely choose.

221 During the recognition phase, the network performs a classification operation to identify which class the input  
222 image belongs to, identifying the one with the highest probability.

223 The values of the filters are initially chosen randomly. They are subsequently improved at each iteration of  
224 the training phase.

225 We estimate that the model's predictions are plausible; in practice, we measure the discrepancy between actual  
226 and predicted values. The error is subsequently processed using the stochastic gradient technique. It is a cyclical  
227 technique consisting of two phases: \* backpropagation; \* updating the gradient value.

228 After propagating the forward phase during training, the outputs are produced to determine the prediction  
229 error compared with the expected ones. This error is used to calculate the gradient of the loss function. The  
230 backward propagation phase then sends the error through the network layers and updates the weights using the  
231 stochastic gradient descent to improve the network's performance on the activity it is trying to learn [29].

---

## 5 IV. Virtual Reality

The virtual environment must have the following characteristics:

? Perception of really being in that world. The use of special equipment amplifies this feeling: software capable of reproducing 3D environments, a virtual reality viewer, integrated audio systems that offer surround support. ? The possibility of interacting with movements of the body, head, and limbs increases the feeling of taking possession of that dimension. For example, cyber-gloves, virtual limbs, joypads, etc., allow the user to touch, move, manipulate, or make virtual objects as if they were real.

Virtual reality technology refers to a computergenerated virtual world [30]. To immerse yourself in this multidimensional world, the user must wear a helmet containing a display that incorporates a sensor to track the movement and position of the wearer (HMD -Head Mounted Display). The sensors detect the position and movement data, which are used to update the image of the virtual world. In this way, the user is projected into this virtual world and can interact with the objects present in it, for example, simulating carrying out usual activities or driving a car or running. By manipulating three variables (space, time, and interaction) and the availability of a graphic interface, it is possible to create a dimension characterized by a strong sense of reality, whereby the subject believes he is in that world and can interact with it.

The relationship between presence and immersion gives cognition in virtual reality. The first term refers to the level of psychological realism that a subject experiences from interacting with the virtual world. From the wise point of view, the second term means the ability of the environment to involve the senses of the subject, isolating it from the stimuli of the real environment [31]. Researchers have been developed methods to dynamically measure human movement in real-time through clothing equipped with sensors that track the full range of actions of specific activities carried out by the wearer. The movement is recorded, digitized, and then sent to the computer, which displays them on the screen, replicating all the actions performed in real-time. directional sensors to transmit electrical signals corresponding to the wearer's movements to a computer. These signals are then converted into digital form, and the resulting data are displayed in the virtual environment. We often use gloves in conjunction with a position and orientation sensor that allows the computer to determine spatial coordinates. They can also be used in conjunction with suits and associated with HMD.

Finally, the viewers called HMD are devices that allow the wearer to immerse themselves in virtual reality. They include a head-mounted stereoscopic display and provide separate images for each eye, a stereo sound, and an ahead motion tracking sensor.

We can distinguish three types of virtual reality:

? Immersive Virtual Reality (RVI)), characterized by the intense sensation of immersion in virtual space thanks to the interaction with the virtual objects present. ? Non-Immersive Virtual Reality, the simulated environment does not stimulate the sense of immersion; it is perceived as the real one. ? Augmented Reality (AR) enriches the objects in the real world with perceptual information produced by the computer through multiple sensory modalities (visual, auditory, tactile).

Below is a brief overview of the state of the art

## 6 V. Background

The functionality of a virtual reality system requires different tools: modeling programs with which objects for the virtual world are created; moreover, to generate dynamics in real-time, simulation systems that follow natural laws are needed to move objects in the virtual world. For example, robust programs for viewing 3D images and spatialized sounds in real-time are also used. In addition, traditional systems used to communicate with a computer are no longer suitable for the multidimensional virtual world.

These interface mechanisms are found in apparel, head-mounted displays (HMDs), and 3D sound systems.

This clothing consists of gloves and suits. Gloves are interactive devices, similar to a typical glove worn on the hand, facilitating tactile sensing and refined motion control. Tactile sensing involves simulating the sense of touch and includes the ability to sense pressure, linear force, temperature, and surface structure. Through the control of the fine movement achieved with sensors, the actions of the user's hands and fingers are detected. These movements are then translated into signals that can be used by a virtual hand to allow the user to interact normally with the objects in the virtual environment. The suit wraps the body with This section reports significant experiences developed based on neural networks combined with virtual reality. Through the analysis of facial expressions, it is possible to understand the emotional state. Neural networks coupled with virtual reality represent a valid platform for analyzing human behavior and effectively treating individuals with autistic disorders. Virtual reality (VR) offers distinct advantages over conventional In this scenario, one of the first applications was that of Beli? [32], who developed a methodology in which the two technologies are used to model complex polycrystalline materials. The neural network is used for generating the grain of the material. It makes the manipulation of the grain easier and the representation of the same very compact. It represents the preliminary phase for realizing the polycrystalline material model for virtual reality. The models perform various system optimization or control activities by simulating reality. To make a realistic model of the observed material, the shape of the grains represented by the neural network must come as close as possible to the observed material. The grains are first generated, then, based on the properties of the observed material, the grain shape optimization process is employed to get closer to the observed sample. With the VR approach, information is obtained that is not accessible with traditional techniques (characterization and analysis techniques). The VR

294 must have the model accompanied by information on the mechanical and electrical properties and on the shape  
295 of the grains. The use of virtual reality made it possible to obtain additional information such as life expectancy,  
296 the diffusion process, or the cracking of the material. VR is used to predict any anomalies produced in the  
297 material; it allows you to view the discrepancies detected concerning the expected model. Finally, the virtual  
298 environment corrected the model to provide better results. The ultimate goal is to obtain the virtual roughness  
299 of the grain as close as possible to the desired one.

300 Finally, the authors used a multilayer neural network (ANN) for speech and motion recognition. The network  
301 architecture and weights have been imported into the Unity3D platform. Without wishing to be a replacement  
302 solution for a dedicated tracking system, it can represent a low-cost method of controlling the gestural execution  
303 of a subject.

### 304 7 ? Collecting training and testing data;

305 ? Extracting features from IMU inputs; ? Building neural network-based classifier models; ? Building test  
306 visualization in Unity.

307 An interesting experience is developed by Nino et al. [33] on a VR mobile system based on Neural Network  
308 to an IMU gesture controller to simulate the feeling of embodiment [34] and presence. The application aims  
309 to demonstrate that combining a sound and motion controller with haptic feedback can improve immersion on  
310 mobile interfaces. The interface recognizes the user's gesture (vocal sound and movement) and displays it during  
311 execution. The visualization is managed through the gesture controller to improve presence and embodiment.  
312 To provide the user with a smooth display of the gesture produced, the authors have created a series of reference  
313 gestures that represent the best possible execution of that gesture. To make the set of these "ideal" gestures,  
314 a motion capture system was used to record the movement and voice produced by a shintaido expert while  
315 performing each gesture [35]. Each ideal gesture is played back for the user while performing the corresponding  
316 gesture. Through the dynamic control of the speed of reproduction of the gesture, the user features a degree of  
317 control over the avatar's movement. When the user has executed his movement, the captured gesture is classified,  
318 and the appropriate response is triggered.

319 In the field of hand gesture recognition, there is an application developed by Fu and Yu [36] in which they  
320 use a neural network for the classification of gestures and, to evaluate the accuracy of recognition in real-time,  
321 they build a Unity scene using the data collected by the IMU. Hand gestures are an alternative to other natural  
322 methods of Communication. The possibility of recognizing the movement and the pose of the hands allows  
323 the observer to understand the intentions of who performs them. Hand gesture recognition is used in many  
324 applications such as sign language recognition, sign recognition for controlling robots, and augmented reality.  
325 The authors used the IMU as a data input source and structured the application into the following phases: VR  
326 Arduino is used as the data input source. It contains an IMU that provides 9 degrees of freedom sensor readings,  
327 including gyroscopes, accelerometers, and magnetometers. It is connected to the PC via USB using Teensy.

328 The sensor data is recorded on the PC as a training data set and used directly in real-time tests.

329 The user can hold the Arduino chip in his hand and perform any hand gesture patterns.

330 A 3D LSTM convolution was used as a neural network to acquire better gestures and obtain high recognition  
331 accuracy.

332 To determine which features work best with the model, they experimented with several combinations of  
333 quaternion and raw sensor inputs. The best performing characteristics were the combination of quaternion,  
334 gyroscope, and accelerometer data. IMU test data from VRduino are transferred directly into Unity via serial  
335 port to visualize the classification.

336 The tests found that the classifier can predict the user's gesture with reasonable accuracy in real-time.

337 Sait and Raza [37] presented a prototype based on an innovative hybrid technology in which are present VR  
338 and ANN models to build a VE environment in which users can meet and chat together and at the same time feel  
339 like they are in a real environment. From a conceptual point of view, the model starts from realworld situations  
340 and then, through the IVR and ANN model, creates a virtual world in which the participants are the managers  
341 of the model, using the Internet as a For the need to have a system design for a realistic visualization, the authors  
342 used the IVR. It is the technology model which allows people to immerse themselves in the artificial environment.

343 ANN is the architecture used to simulate the image and express the user's emotion in the VE. Users' voices,  
344 videos, and images are input to train the ANN. Those data are also used for security purposes to check the users'  
345 identities.

346 The Internet connects people spread over different points into a single point of contact on the web. The ANN  
347 network is trained on the Internet with a sample of users. The Internet allows scalability for a large number of  
348 users and ensures reliable transport of a large amount of multimedia data.

349 The database must contain a large number of quality video samples and possess a large number of facial  
350 recognition expressions to store the video and audio data of the participants.

351 The model requires two implementation steps. The first is for the interface, and the user inserts his data into  
352 the model, followed by the loading of their image, voice. These data will be stored in the audiovisual database  
353 to the interface to configure the user with the model. Later, when the user enters the system, the interface will  
354 check their identity with the stored data. Finally, the interface must be designed using 3D graphics software [38].

---

355 In the second phase, the IVR will provide the chosen environment to those users participating in the network,  
356 and the ANN will provide simulated images of the users.

357 As noted above, there is a growing need for non-experts to understand how deep learning works. For their  
358 complexity, neural networks are seen as black boxes. New techniques are experimented with to make the  
359 understanding of their functionality as accessible as possible.

360 In this context, Meissler et al. [39] have developed a technique for visualizing convolutional networks in virtual  
361 reality. Their solution is mainly aimed at new designers and aims to provide them with a basic understanding  
362 of the functionality of networks. The solutions currently available are essentially usable either by developers of  
363 deep learning systems or by those who already have detailed knowledge of deep learning processes or by those  
364 who generally have an interest in interactive visualization but are not attracted by the representation capabilities  
365 of virtual reality.

366 Their solution stems from using the advantages of virtual reality in visualization compared to traditional 2D  
367 or 3D tools available on a standard desktop screen. Their approach explores the graphic effects produced by  
368 the immersion feature. The visualization is intended to illustrate the structure and functionality of the network.  
369 For example, the user within the Unity platform can select different inputs, modify the functional structure by  
370 inserting or removing a layer and obtain their classification by CNN.

371 The network model used is LeNet5 [40], whose structure is as follows: ? 60000 trainable parameters. The  
372 model was defined in Keras [41] using PYTHON.

373 The convolutional and pooling layers were displayed as boxes whose dimensions were based on the layer's size.  
374 This type of representation should allow the user to see the structure of the model and the dimensional change  
375 of the data. Feature maps of each convolutional and pooling layer are rendered 2D images with matplotlib using  
376 the gray colormap.

377 The experiments carried out on the participants who did not know the topic confirmed the model's validity.  
378 Some of them declared that the application was completely self-explanatory. The experiments wanted to  
379 demonstrate that virtual reality compared to traditional means of learning, does not distract people but, on  
380 the contrary, allows them to focus more on the subject and even longer due to the involvement produced by the  
381 virtual environment.

## 382 8 VI. Methodology

383 Our solution is based on a VR platform where users can develop deep convolutional neural network models for  
384 image classification. This type of approach allows verifying the components of a network and becomes a good  
385 training tool for professionals who want to design deep neural networks.

386 In our work, we want to show the opportunity to use the Unity platform (2019) to create a virtual reality  
387 environment. We believe it offers some advantages not found in various visualization systems in 2D and 3D  
388 desktop screens. Most of the existing platforms do not provide, for inexperienced users, simple interfaces through  
389 which it is possible to vary the network architecture and adjust the model parameters. For example, the VR  
390 platform allows the user to define the network architecture by specifying the sequence of the layers. The user  
391 can interactively verify its operation and decide based on the results obtained by modifying it. We have used  
392 this environment to verify the functioning and possibly improve the prediction level of a 1 shows the network  
393 architecture graph, while figures 2 and 3 show, for example, the information relating to the selected layers Conv2D  
394 and Dense obtained by clicking directly on the chart. This information includes: the type of the layer; the name  
395 of the layer, whether the layer is trainable; what the data type is. For Convolutional layers: the number of  
396 filters; the kernel size; the strides; padding; data format and dilation rate. The activation function, whether  
397 bias is used, and how the kernels and (if applied) biases are initialized. Each layer has its own unique set of  
398 characteristics. ONNX, which stands for Open Neural Network Exchange, is an open standard that allows you  
399 to transfer machine learning models from different frameworks to ONNX. This interoperability enables you to  
400 move between various machine learning frameworks quickly. In addition, ONNX supports all popular machine  
401 learning frameworks, including Keras, TensorFlow, PyTorch, and XGBoost.

402 ONNX allows you to have a standard graphical representation for various frameworks. The ONNX  
403 graph represents the model through different computational nodes and can be visualized using appropriate  
404 representation tools. The network model in ONNX format is imported into Unity as a temp file (Fig. ??), whose  
405 structure will appear in the following graph (Fig. 5).

## 406 9 Fig. 4

407 To perform the conversion, we used a script. An extract of which is represented by the following code:

408 Within Unity, we used the Barracuda package developed by Unity Labs to develop our application. It is a  
409 cross-platform neural network inference engine. Pretrained neural networks can be imported and run in Unity.  
410 In addition, it supports neural networks trained with PyTorch, TensorFlow, Keras, and Caffe. We executed the  
411 following steps:

412 To transfer the network, it is necessary to transform its format, as Barracuda is based on the open-source data  
413 exchange format ONNX. As a next step, we need to create the inference engine. The core engine interface in

414 Barracuda is called worker. IWorker breaks down the model into executable tasks and schedules them on GPU  
 415 or CPU.

416 The scripts are different depending on whether you use a GPU or a CPU. We can create a Worker from the  
 417 WorkerFactory. We must specify a backend and a loaded model. In our case, to improve the accuracy of the  
 418 network, we modified the structure by inserting an additional convolutional layer obtaining a value of 99%.

419 **10 VII. Discussion**

420 **11 VIII. Conclusion**

421 We have analyzed the most common visualization techniques used to represent neural networks, a valuable tool for  
 422 their development. Still, we have found that some are sophisticated tools that are difficult to use for inexperienced  
 423 designers. Therefore, we have highlighted how to apply Virtual Reality to construct deep learning models. This  
 424 type of representation, in fact, through the visualization of technical information on the network structure and  
 425 on the data flow, represents an effective tool both for improving the forecasting capacity of neural networks and  
 426 for training activities for those who do not have a profound knowledge of deep learning systems.

427 We have experimented with applying it to a pre-trained neural network to recognize human activities (HAR).  
 428 Still, we believe that it can effectively design all neural networks for image classification or object detection.

429 Our approach shows how the VR technique can effectively design deep learning applications. Our platform  
 430 represents a valuable tool for developers with little knowledge of machine learning, allowing rapid design. It is  
 431 an interactive technique also valid as a learning tool. Compared to existing visualization techniques, Virtual  
 432 Reality is a potential highly developed for neural network applications. Above all, it has proved to be suitable  
 for classifying images in different scientific sectors. We can assume that it is applicable for any deep learning.

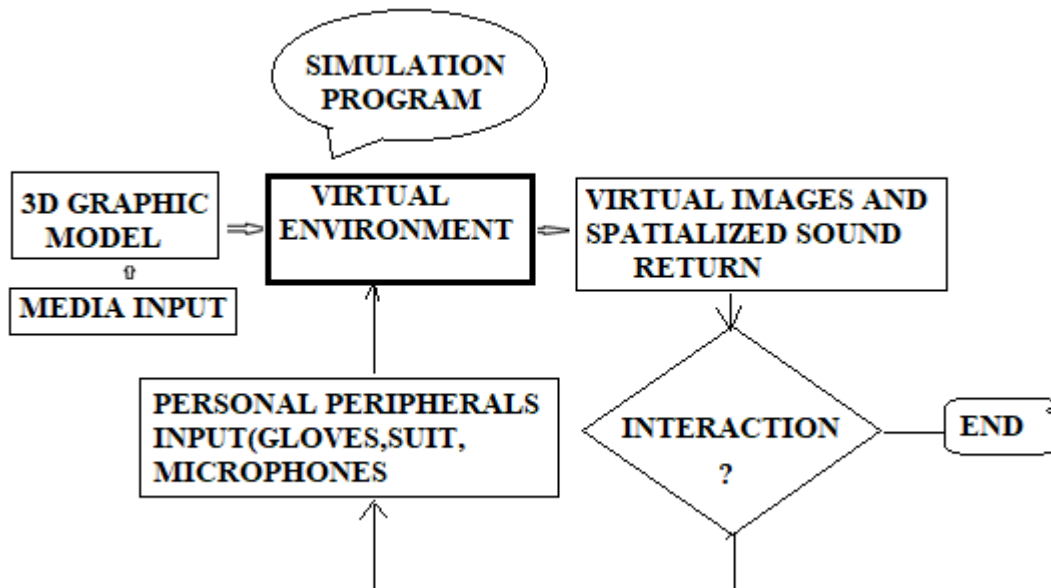


Figure 1:

433 1 2  
 434

<sup>1</sup>© 2022 Global Journals

<sup>2</sup>© 2022 Global Journals Neural Network Design using a Virtual Reality Platform



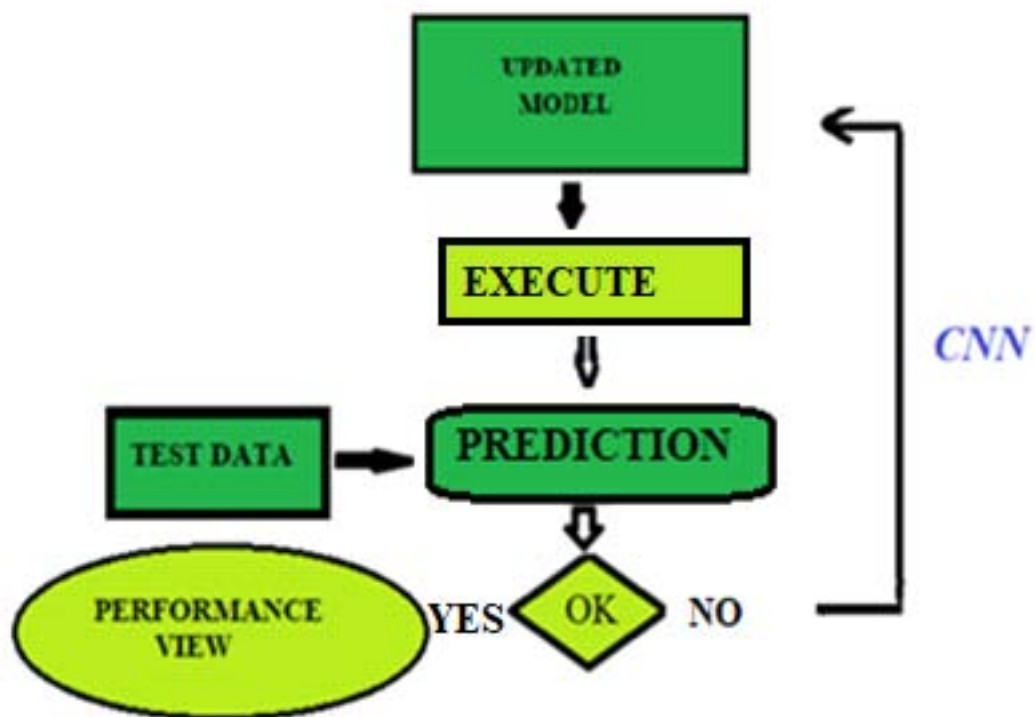


Figure 2:



**NODE PROPERTIES** ✕

bias_initializer	Zeros()
data_format	channels_last
dilation_rate	1, 1
filters	128
kernel_initializer	VarianceScaling(scale: 1, mode: "fan_avg", distribution: "uniform", seed: null)
kernel_size	2, 2
padding	valid
strides	1, 1
trainable	true *
use_bias	true *

**INPUTS**

input	name: input
kernel	name: conv2d_1/kernel:0 *
bias	name: conv2d_1/bias:0 *

**OUTPUTS**

output	name: conv2d_1
--------	----------------

2

Figure 4: ? 2

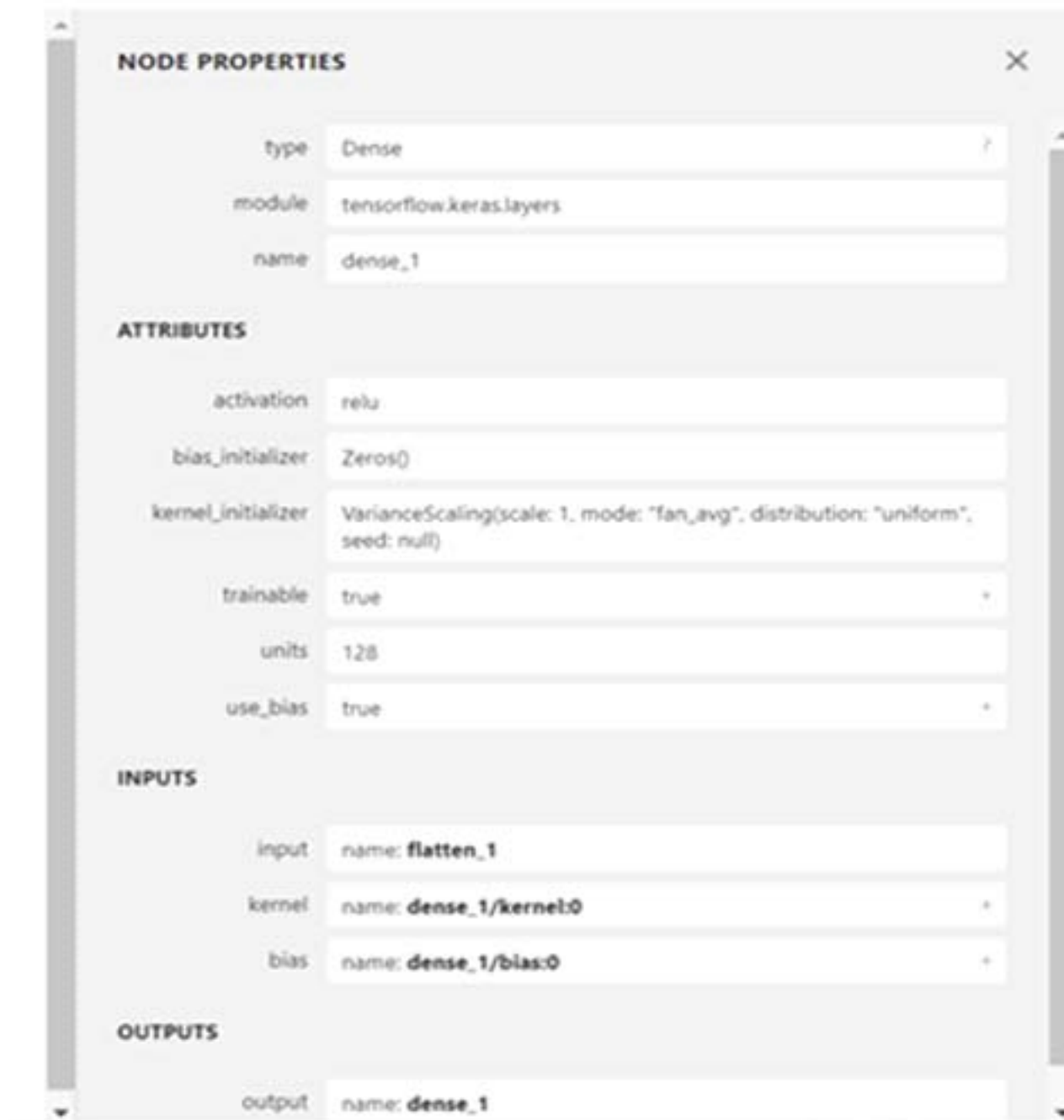


Figure 5:

```

ported Object
model.h5
Open imported NN model as temp file

Source: ONNX
Version: 9
Producer Name: tf2onnx v1.9.2

Inputs (1)
conv2d_input shape: (n, h, w, c)

Outputs (1)
dense_2 shape: (n, h, w, c)

Memories (0)

Layers (12 using 738.950 embedded weights)
Conv2D name sequential/conv2d/BiasAdd, activation ReLU, inputs [conv2d_input], pad [0,0,0,0], weights [sequential/conv2d/BiasAdd/0/K (n,2, h,2, w,1, c,128), sequential/conv2d/BiasAdd/0/B (n,1, h,1, w,1, c,128)]
MaxPool2D name sequential/max_pooling2d/MaxPool, inputs [sequential/conv2d/BiasAdd], pad [0,0,0,0], stride [2,2], pool [2,2]
Reshape name sequential/flatten/Reshape, inputs [sequential/max_pooling2d/MaxPool], pool [1,1,-1,1,1,1,5632]
Dense name sequential/dense/MatMul, inputs [sequential/flatten/Reshape], weights [W (n,5632, h,1, w,1, c,128), B (n,1, h,1, w,1, c,128)]
Add name sequential/dense/BiasAdd, inputs [sequential/dense/MatMul,c,sequential/dense/BiasAdd/for_sequential/dense/BiasAdd], axis 1
Activation name sequential/dense/ReLU, activation ReLU, inputs [sequential/dense/BiasAdd]
Dense name sequential/dense_1/MatMul, inputs [sequential/dense/ReLU], weights [W (n,128, h,1, w,1, c,128), B (n,1, h,1, w,1, c,128)]
Add name sequential/dense_1/BiasAdd, inputs [sequential/dense_1/MatMul,c,sequential/dense_1/BiasAdd/for_sequential/dense_1/BiasAdd], axis 1

Constants (3 using 262 weights)
Load name c, sequential/dense_2/BiasAdd/for_sequential/dense_2/BiasAdd, weights [(n,1, h,1, w,1, c,8)]
Load name c, sequential/dense_1/BiasAdd/for_sequential/dense_1/BiasAdd, weights [(n,1, h,1, w,1, c,128)]
Load name c, sequential/dense/BiasAdd/for_sequential/dense/BiasAdd, weights [(n,1, h,1, w,1, c,128)]

```

1

Figure 6: Fig. 1 :?

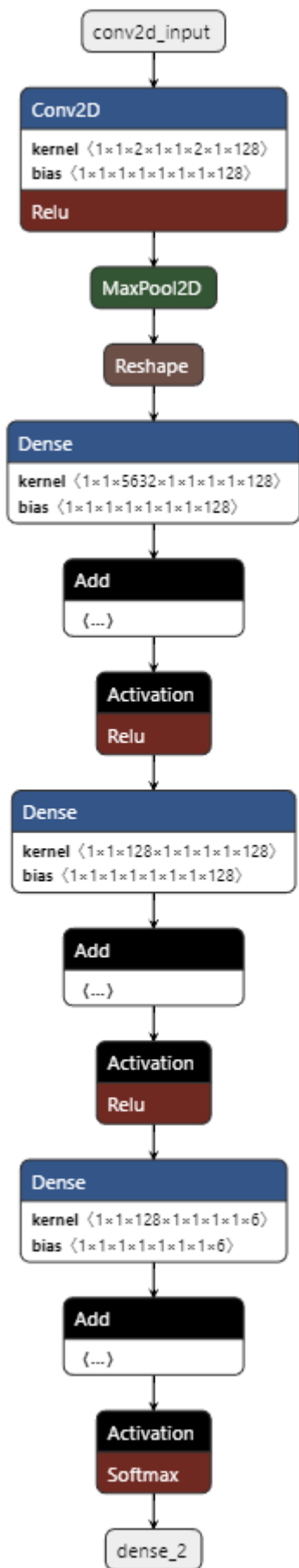


Figure 7:  
14

---

## 4

*[Note: During the training phase, these weights are progressively calibrated so that more and more, the network's output in the face of a specific input approaches the desired one. This phase is implemented by comparing the initial classification with what is suggested by learning. We use an algorithm called Backpropagation is used to train neural networks. It compares the result obtained from a network with the output you want to have. We use the difference between the two results to change the weights of the connections between the network layers starting from the output layer. Then we proceed backward by modifying the weights of the hidden layers and finally those of the input layers. To do this, we use a function called cost appropriate to the problem we want to solve. A part of the data prepared is used for the training phase while the data for the test and validation phase are kept separate. Since pre-trained networks are available, we can use them to classify new objects using the transfer learning technique. 5) The process of evaluating new images using a neural network to make decisions is called inference. This step can collect additional test data used as training data for future iterations.]*

Figure 8: 4 )





## 1 Acknowledgments

- This work was developed as an integral component of the project: "An integrated System for indoor people localization, tracking, and monitoring" supported by the Italian MIUR under GRANT PON Research and Innovation 2014-2020 Project Code C35E190000200001, AIM1839112-1: Technologies for living environments.
- [Chow ()] 'A Cost-Effective 3D interaction Approach for Immersive Virtual Reality'. Yang -Wai Chow . *The International Journal of Recent Trends in Engineering* 2009. 1 (1) p. .
- [Lee et al. ()] 'A Study on Immersion and VR Sickness in Walking Interaction for Immersive Virtual Reality Applications'. J Lee , M Kim , J Kim . *Symmetry* 2017. 9 (5) p. 78.
- [Sait and Raza ()] 'A Virtual Environment Using Virtual Reality and Artificial Neural Network'. A R W Sait , M N Raza . *IJACSA International Journal of Advanced Computer Science and Applications* 2011. 2 (12) .
- [Kahng et al. ()] 'Activis: Visual exploration of industry-scale deep neural network models'. M Kahng , P Y Andrews , A Kalro , D H P Chau . *IEEE Trans. Visual. Comput. Graph* 2017. 24 p. .
- [Bibbò and Carotenuto] 'An integrated system for indoor people localization, tracking, and monitoring'. L Bibbò , R Carotenuto . *Journal of International Scientific Publications* 1314- 7269. 15 p. 2021. (Materials, Methods & Technologies)
- [Arnold et al. ()] 'An Introduction to Deep Learning'. L Arnold , S Rebecchi , S Chevallier , H Paugam-Moisy . *European Symposium on Artificial Neural Networks (ESANN)* 2011.
- [Li et al. ()] 'Brief Introduction of Back Propagation (BP) Neural Network Algorithm and Its Improvement'. J Li , J Cheng , J Shi , F Huang . *Advances in Computer Science and Information Engineering* 2012. 169 p. 558.
- [Chattopadhyay et al. ()] A Chattopadhyay , P Manupriya , A Sarkar , N B Vineeth . arXiv:1902.023024. *Neural Network Attributions: A Causal Perspective*, 2019.
- [Biocca and Delaney ()] *Communication in the age of virtual reality*, F Biocca , B Delaney . 1995. Lawrence Erlbaum Associates, Inc. p. . (Immersive virtual reality technology)
- [Rawat and Wang ()] 'Deep convolutional neural networks for image classification: A comprehensive review'. W Rawat , Z Wang . *Neural computation* 2017. 29 (9) .
- [Lecun et al. ()] 'Deep learning'. Y Lecun , Y Bengio , G Hinton . *Nature* 2015. 521 p. .
- [Goodfellow et al. ()] *Deep Learning*, I Goodfellow , Y Bengio , A Courville . 2016. MIT Press. p. 32.
- [Pezzotti et al. ()] 'Deepeyes: Progressive visual analytics for designing deep neural networks'. N Pezzotti , T Höllt , J Van Gemert , B Lelieveldt , E Eisemann , A Vilanova . *IEEE Trans. Visual. Comput. Graph* 2018. 24 (1) p. .
- [Liu et al. ()] 'Deeptracker: Visualizing the training process of convolutional neural networks'. D Liu , W Cui , K Jin , Y Guo , H Qu . *ACM Transactions on Intelligent Systems and Technology* 2018. 10 (1) p. .
- [Nino et al. ()] 'Enhancing Mobile VR Immersion: A Multimodal System of Neural Networks Approach to An IMU Gesture Controller'. Juan Nino , Jocelyne Kiss , Geoffrey Edwards , Ernesto Morales , Sherezada Ochoa , Bruno Bernier . *The Engineering Reality of Virtual Reality, IS&T International Symposium on Electronic Imaging*, 2019. p. .
- [Sattarzadeh et al.] 'Explaining Convolutional Neural Networks through Attribution-Based Input Sampling and Block-Wise Feature Aggregation'. S Sattarzadeh , M Sudhakar , A Lem , N Mehryar , J Plataniotis , H Jang , Y Kim , S Jeong , K Lee , Bae . arXiv:2010.00672. *Computer Vision and Pattern Recognition*,
- [Mordvintsev and Schubert ()] 'Feature Visualization'. C , Olah A Mordvintsev , L Schubert . *Distill* 2017. 2 (11) .
- [Lecun et al. ()] 'Gradient-based learning applied to document recognition'. Y Lecun , L Bottou , Y Bengio , P Haffner . *Proceedings of the IEEE* 1998. 86 (11) p. .
- [Spanlang et al. ()] 'How to build an embodiment lab: achieving body representation illusions in virtual reality'. B Spanlang , J.-M Normand , D Borland , K Kilteni , E Giannopoulos , A Pomes , M Gonzalez-Franco , D Perez-Marcos , J Arroyo-Palacios , X N Muncunill . *Frontiers in Robotics and AI* 2014. 1 p. 9.
- [Weber et al. ()] 'How to Get There When You Are There Already? Defining Presence in Virtual Reality and the Importance of Perceived Realism'. S Weber , D Weibel , F W Mast . *Frontiers in Psychology* 2021. 12 p. 1538.
- [Psotka ()] 'Immersive training systems: Virtual reality and education and training'. J Psotka . *Instructional Science* 1995. 23 (5-6) p. .
- [Chollet ()] *Keras*, F Chollet . <https://keras.io> 2015.
- [Kocaoglu et al. ()] *Learning causal implicit generative models with adversarial training*, M Kocaoglu , C Snyder , A G Dimakis , S Causalgan , Vishwanath . arXiv:1709.02023. 2017.

- 490 [Montavon et al. ()] ‘Methods for interpreting and understanding deep neural networks’. G Montavon , W Samek  
491 , K R Muller . *Digital Signal Processing* 2017.
- 492 [Nguyen et al. ()] *Multifaceted feature visualization: Uncovering the different types of features learned by each*  
493 *neuron in deep neural networks*, A Nguyen , J Yosinski , J Clune . arXiv:1602.03616. 2016.
- 494 [Beli ()] ‘Neural Network Modelling and Virtual Reality’. I Beli . *The Thousand Faces of Virtual Reality cap*,  
495 2014. 7.
- 496 [Benardos and Vosniakos ()] ‘Optimizing feedforward artificial neural network architecture’. P G Benardos , G  
497 C Vosniakos . *Engineering Applications of Artificial Intelligence* 2007. 20 (3) p. .
- 498 [Fu and Yu ()] *Real-time Gesture Pattern Classification with IMU Data*, A Fu , Y Yu . 2017.
- 499 [Chen et al. ()] ‘Tensorview: visualizing the training of convolutional neural network using paraview’. X Chen ,  
500 Q Guan , X Liang , L T Lo , S Su , T Estrada , J Ahrens . *1st Workshop on Distributed Infrastructures for*  
501 *Deep Learning DIDL '17*, 2017. ACM Press. p. .
- 502 [Cruz-Neira et al.] ‘The cave: audiovisual experience automatic virtual environment’. C Cruz-Neira , D J Sandin  
503 , T A Defanti , R V Kenyon , J C Hart . *Communications of the ACM* 35 (6) p. .
- 504 [Franklin ()] ‘The Meaning of Tenshingoso’. D Franklin . *Journal of the U.S. Shintaido Movement Issue* 2001.  
505 10.
- 506 [McInnes et al. ()] *Umap: Uniform manifold approximation and projection for dimension reduction*, L McInnes  
507 , J Healy , J Melville . arXiv:1802.03426. 2018.
- 508 [Shalev-Shwartz and Ben-David ()] *Understanding Machine Learning: From Theory to Algorithms*, Shai Shalev-  
509 Shwartz , Shai Ben-David . 2014. Cambridge University Press.
- 510 [Yosinski et al. ()] *Understanding neural networks through deep visualization*, J Yosinski , J Clune , A Nguyen ,  
511 T Fuchs , H Lipson . arXiv:1506.06579. 2015.
- 512 [Meissle et al. ()] ‘Using Visualization of Convolutional Neural Networks in Virtual Reality for Machine Learning  
513 Newcomers’. N Meissle , A Wohlan , N Hochgeschwender , A Schreibe . *IEEE International Conference on*  
514 *Artificial Intelligence and Virtual Reality (AIVR)*, 2019.
- 515 [Abulrub et al. ()] ‘Virtual reality in engineering education: The future for creative learning’. A G Abulrub , A  
516 N Attridge , M A Williams . *IEEE 2011*. 2011. p. .
- 517 [Peters et al. ()] *Visualising the Training Process of Convolutional Neural Networks for Non-Experts*, M Peters  
518 , L Kempen , M Nauta , C Seifert . 2941. 2019. 108. (CEUR-WS.org)
- 519 [Bock and Schreiber ()] ‘Visualization of neural networks in virtual reality using UNREAL ENGINE’. M Bock ,  
520 A Schreiber . *24th ACM Symposium on Virtual Reality Software and Technology, ser. VRST '18, ACM*, 2018.  
521 2018. 132 p. .
- 522 [Erhan et al. ()] *Visualizing higher-layer features of a deep network*, D Erhan , Y Bengio , A Courville , P Vincent  
523 . 2009. p. 4323. Dept. IRO, Universite de Montreal (Tech)
- 524 [Sutton] *What Is Virtual Reality Therapy? The Future of Psychology*, Jeremy Sutton . p. 2020.