A Computational Cognitive Model of Human Memory Based on Invertible Neural Networks

Zheyuan Zhang
College of Information and Computer Sciences
University of Massachusetts, Amherst
Amherst, MA
zheyuanzhang@umass.edu

Abstract

Cognitive modeling is a prerequisite for an intelligent agent to be more like humans or other animals, and memory is the basis for higher mental activities such as thinking, generating emotions, and imagining. This paper presents a computational model of memory to simulate how we remember and recall things. It demonstrates the feasibility of using neural networks to encode information stored in the memory. The computational model is based on the multi-store model of memory, which divides the memory into a sensory register, a short-term store, and a long-term store. The model accomplishes the process of recovering memory traces by using invertible neural networks (INNs). Furthermore, the paper established a bridge between artificial intelligence and psychology, in which psychology can inspire and advance AI while AI can be used to explain psychology in a computational manner.

1 Introduction

The brain remains a mystery to humans. At present, we are unable to create a functionally similar artificial brain because we do not understand well enough how the brain achieves cognition. The brain is very complex, so it is hard to model, but it is feasible to model or simulate some functionalities independently. For example, computer vision simulates visual perception, and natural language processing simulates the understanding of human language.

However, there is little research on how to model human memory. We emphasize the study of perception because human perception is very strong, while human memory is much less capable than computers. The initial motivation of an artificial intelligence advancement can be categorized into two classes. One class aims to surpass humans, while another class mimics humans or other animals. For example, a search algorithm like uniform-cost search can be applied to a state-based problem and guarantee an optimal solution. Another example could be a minimax adversarial search applied to a state-based game. The two examples shown above can generally perform better than humans, but the exhaustive approach does not make the machine intelligent. This is also the reason that most robots cannot have good sales in the market because their intelligence cannot be compared to actual animals. Memory is responsible for encoding, storing and recalling information. Therefore, a computational memory model is a prerequisite to high-level AI that approaches natural intelligence to make the machine more like humans and other animals.

The way computers today store information is very different from the brain. The primary method for a computer to remember an image is to store every pixel with color information in a multi-dimensional array. It is different for a human to store information. What we remember in our memory are features of an image. However, we cannot consider the memory to be a big list of features. According to the

Preprint. Under review.

multi-store model of memory[2], there are three stores, which are sensory register, short-term store, and long-term store. This paper will simulate the memory based on this model.

In recent years, new deep learning models like ResNet[11], Transformer[16], BERT[7], ViT[8], MAE[10] has been consistently achieving better results in CV and NLP. These state-of-the-art models' high performances on datasets in multiple tasks show that current neural network models can extract high-quality features in images and texts. Feature extraction will play an important role in simulating human memory. The majority of the following paper will be organized into five sections: Information Encoding, System Structure, Recovering Memory Trace, Bridging Psychology and Conclusions and Future Work.

2 Information Encoding

2.1 Method

The resolution of the human eye is at least 576 megapixels[4] which is equivalent to 576 million pixels, so a human cannot record every pixel in our memory. Instead, the memory encodes raw sensory data into features stored in the brain. We can recognize the encoder in our memory as a well-trained neural network that can extract features of everything we see, hear, smell, taste, and touch (only visual information processing are demonstrated in this paper). Therefore we can use a pre-trained neural network to be our encoder and the implementation is in the following subsection.

2.2 Implementation

An invertible neural network needs to be used to recover original information from features, which will be discussed in the Recovering Memory Trace section. For the simplicity of the code in this section, it uses a pre-trained ResNet50 model to demonstrate how it works. Also, this paper is not intended to have the best simulation but to provide an approach to reverse engineer the memory and model it. However, using other state-of-the-art models for feature extraction may lead to better simulation.

In order to extract features instead of doing classification, the last 1000-d fully connected classification layer is removed. The last average pooling layer outputs the feature with the shape $[1, 2048, 1, 1]$. After that, the feature is squeezed to $[2048]$ for more straightforward computation later.

2.3 Experiments

Here are 6 images of cats and dogs from ImageNet[6] which will be used to test whether the model can be used to extract features. All images are transformed by resizing to 224×224 and normalizing with the mean and standard deviation of ImageNet ($mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]$) shown in Figure 1.

(a) $cat - 1$ (b) $cat - 2$ (c) $cat - 3$ (d) $dog - 1$ (e) $dog - 2$ (f) $dog - 3$

Figure 1: Test images

The squeezed output features of each image from the ResNet50 model are shown in the Table 1.

Table 1: Image features

Image	Feature
$cat-1$	[0.19761689 0.2173356 0. ... 0.38727978 0.2638614 0.36926955]
$cat-2$	[0.19163685 0.16676304 0.08433288 ... 0.23582394 0.19334112 0.07674974]
$cat-3$	[0.21620344 0.32265875 0.08948374 ... 0.11365888 0.09229241 0.15956782]
$dog-1$	[0.15818311 1.0075512 0.7031254 ... 0.3126888 0.14855877 0.14108977]
$dog-2$	[0.83112234 0.43695694 0.86641484 ... 0.13985324 0.2446276 0.13236925]
$dog-3$	[0.03144252 0.57331145 0.46354374 ... 0.12139889 0.0799183 0.16189402]

These features are examples of encoded information stored in the artificial human memory. All the feature tensors are 2048 long. Even if we use a camera with a human-eye resolution, we can still downsample and resize the image to 224×224 and follow the same process to extract features. Compared to 576 million pixels, the information size is much reduced. Moreover, these features contain information that is closer in the form of what human stores in our memory. For instance, we know two cats are the same animal even if they look different in color, size, and background.

Therefore, the feature of an object compared to another object in a different category should be more different than an object compared to another in the same category. The following contrastive experiments are conducted to observe what kind of information is encoded in these long tensors. The distance measures are cosine distance and L^2 norm (Euclidean distance).

$$cosine\,distance = 1 - \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} \quad and \quad L^2\,norm = \|\mathbf{u} - \mathbf{v}\| \tag{1}$$

Table 2 shows the differences between the feature of $cat-1$ and features of other 5 animals:

Table 2: Feature comparison of $cat-1$

Image	Cosine distance	L^2-norm
$cat-2$	0.2010	15.8703
$cat-3$	0.1809	15.2216
$dog-1$	0.4120	22.7588
$dog-2$	0.4510	25.0965
$dog-3$	0.3998	22.2742

Table 3 shows the differences between the feature of $dog-1$ and features of other 5 animals:

Table 3: Feature comparison of $dog-1$

Image	Cosine distance	L^2-norm
$dog-2$	0.2144	16.5805
$dog-3$	0.2533	16.5054
$cat-1$	0.4120	22.7588
$cat-2$	0.4001	20.0041
$cat-3$	0.4470	22.0415

The above results show that the cat is closer to other cats and the dog is closer to other dogs in the feature space, which means the neural network accurately captures the features, which is consistent with our common sense.

3 System Structure

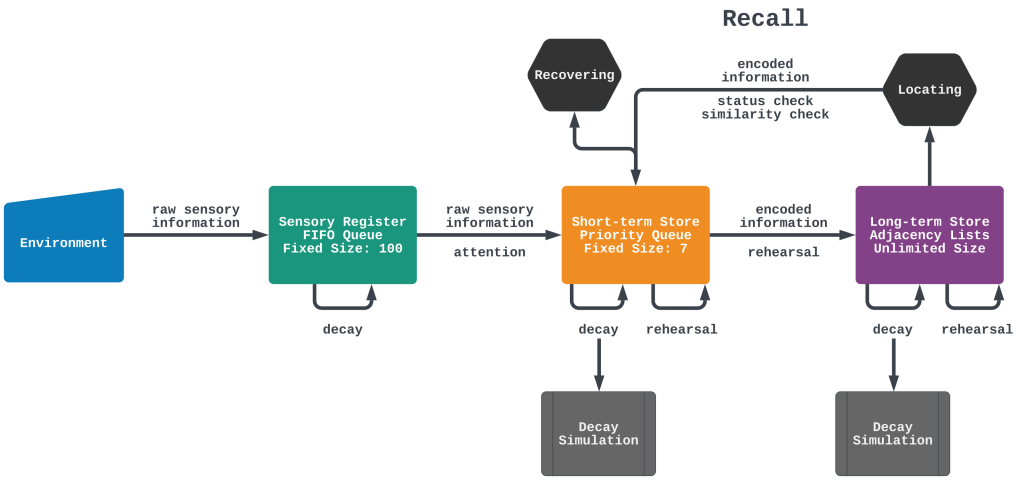

Figure 2: A computational system structure of human memory based on the multi-store model

3.1 Decay Simulation

Although it looks pretty natural for anything in the world to decay biologically, it is computationally inefficient for a computer to simulate the decay. One possible solution is to treat every information unit as an independent process with its decay process. This may work in the simulation of short-term memory, which only has several units of information. However, there are countless units of information in the long-term store. Apparently, it is impossible to implement multi-processing in this component. Also, it is unnatural to simulate the decay process by multi-processing. The second solution is to set a central coordinator to refresh all the information units in each memory component. This will require the machine to iterate every unit for every millisecond if we set our smallest unit of time for decay to be one millisecond. Needless to say, this requires considerable computation power to perform. For simulation rather than cloning, we can make a few changes to the original model, discussed in the following subsections.

Another inefficiency is clocking. Any matter in the world decays naturally but to simulate decay in a machine, the simulation program has to make function calls to the system clock and execute conditional statements to schedule the decay. As we cannot avoid function calls to the system clock, the smallest unit of time for decay needs to be extended to reduce computational cost. It may not have a noticeable effect on the simulation as a human cannot realize a memory trace decayed in the last second or last millisecond. Additionally, there are two possible implementations of decay. One is to add a negative decay constant for every unit of time. Another is to multiply a decay rate ($0 < r < 1$). However, both methods are viable for simulation after proper number setups. Minus-decay is used to increase inefficiency as multiplication usually is slower than addition.

To simulate realistic decay, not only the strength of the memory trace will be decreased, but the information in the memory trace will also be decayed. It should be noted that the information decay happens when the strength is lower than 0.5. The information decay algorithm shown below decays a percentage of information in a feature. The decayed part can be random numbers between 0 to 1 or simply be 0.

Algorithm 1 Information decay

Input: Feature F as list, decay constant D as float
$L \leftarrow$ length of F
$N \leftarrow int(L \times D)$
$idxes \leftarrow$ randomly sample N indexes from range 0 to L without replacement
for each index i of $idxes$ **do**
    $F[i] \leftarrow$ a random floating-point number between 0 and 1 OR 0
end for
Return F

3.2 Sensory Register

Sensory memory is the beginning stage of the human memory which holds all incoming information for a very short period of time [9, p. 121]. The data structure of the sensory register is set to be a FIFO queue with a fixed size of 100 which means it can hold up to 100 units of information. There is no explicit decay constant in the sensory register because it constantly receives information and the duration is very short. Therefore with this implementation, the sensory register pops out an information unit when a new information unit pushes into the queue.

The complete process of how the sensory register works is as follows. Firstly, the brain makes an action to perceive sensory information from the environment, including visual, auditory, haptic, olfactory and gustatory information. Only visual information will be used in this simulation example since we use the ResNet50 model to extract features. After the sensory information is captured, it will be passed into memory with raw data, information type and attention ($0 < a \leq 1$). The information unit with attention greater than 0.5 will be transferred to the short-term store, and the rest of the information units will stay in the sensory register. Here, the information is not going to be encoded. In the end, which is a very short period, old information units will be deleted forever in the sensory register.

3.3 Short-term Store

The short-term store is the second component of the memory system, which holds several information units for several seconds. The exact number of information units has some controversies where some scholars believe the maximum capacity is 7 ± 2 units [13] and some scholars argue that the number is 4 [5]. In this part, I use 7 units as the maximum capacity. The data structure of the short-term store is set to be a priority queue. The information units will be popped off from the priority queue if the strength of the memory trace is decayed to be less or equal to 0. Also, if the number of total information units in the short-term store exceeds 7, the trace with the lowest strength will be popped off. This operation can be done in $O(1)$ as a priority queue is used.

The information is encoded before entering the short-term store by extracting features from the ResNet50 pre-trained model. A clock will update every memory trace with decayed strengths for every second to simulate decay. A new incoming trace has a strength ($0.5 < s \leq 1$) equal to the attention factor. The decay constant is 0.05, so it will take 10 to 20 seconds to remove an information unit.

Rehearsal is a vital control process in short-term memory. In this model, rehearsal is a simple operation that adds a constant number to the strength of the rehearsed memory trace. This can maintain the information from being removed by decaying. Also, rehearsal can transfer the information unit from the short-term store to the long-term store if the strength exceeds 1, which means an information unit can be transferred to a long-term store if rehearsed at least once.

3.4 Long-term Store

The long-term store is the third and the last component, and it can hold a large amount of information for unlimited time. All memory traces are stored in a list of information units. Additionally, the long-term store is a large graph of information units where each edge represents a connection between two memory traces. A connection is established when two traces have the following conditions: high similarity in shallow representations, high similarity in deep representations, sequential order

in time. For the data structure of the graph, we can use adjacency lists to represent connectivity in memory-efficient way [14].

The decay process is very slow in this component, so the decay constant is 0.0005, and the clock will update all memory traces every day. Thus, it will take 1000 to 2000 days for an information unit in the long-term store to decay completely. Rehearsal also exists in the long-term store, which will perform similar functionality as described in the short-term store. Basically, it will make the information stay longer. As noted earlier, some memory traces are connected, so the rehearsal will also increase the strength of neighboring information units.

Recall is one of the main functionalities of memory. The recall process has two stages. The first stage is locating the information unit in the long-term store. The brain will generate a feature to find a match in the long-term store based on the distance measure. This guarantees to find a match as it will always choose the one with the lowest distance in $O(N)$. The information unit has an entry that indicates this memory trace's status. The status can be valid or invalid. If the matched information unit's status is valid (status check) and has a distance smaller than a threshold (similarity check), the memory trace will be transferred into the short-term store along with its neighbors. It is recursive, which means it will consistently transfer connected information as long as the previous information has a strength greater than a threshold. This process also acts as a rehearsal process to increase the strength of recalled memory traces. The second stage is recovering the original information from the encoded feature in the unit, which is done in the short-term store, also known as the working memory. The recovering process is demonstrated in the next section.

4 Recovering Memory Trace

The raw data are decoded into features, which is the memory trace, stored in the long-term memory. The previous section mentioned that the memory trace located will be transferred to working memory for recovering. The recovering process will revert the encoded information to the resized image.

Invertible neural networks (INNs) can be used for recovering images from hidden representations. INNs are bijective, which has a forward mapping from input to output and an inverse mapping from output to input [1]. In this paper, I use i-RevNet [12] as the model to perform recovering because information encoded is of type image. The encoded information outputs from the last layer of the network before the average pooling layer because the pooling layer will cause information loss, leading to a mapping that is not invertible. The shape of the feature is $[6, 3072, 7, 7]$. It will then flattened to $[903168]$ to be easier to apply the decay algorithm. After that, it will be reshaped to the original size $[6, 3072, 7, 7]$ for inverting.

The recovering process of encoded information depends on noises in the feature. Figure 3 and Figure 4 below shows the recovering images from features with different decay percentages which use zeros as decayed information. If the decay percentage is 0, the image can be recovered totally. However, we can see that while the decay percentage increases, the recovered image is becoming noisier. This phenomenon is cognitively plausible because we can immediately recall most of an image when we see an image. However, if the memory trace is not rehearsed for a long time, we may only recall the basic outline of that image.

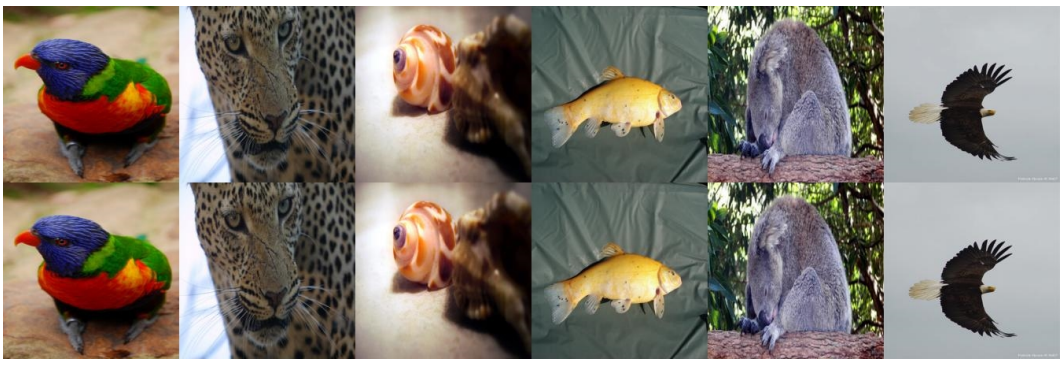

Figure 3: First row: input images. Second row: recovered images from features with no decay.

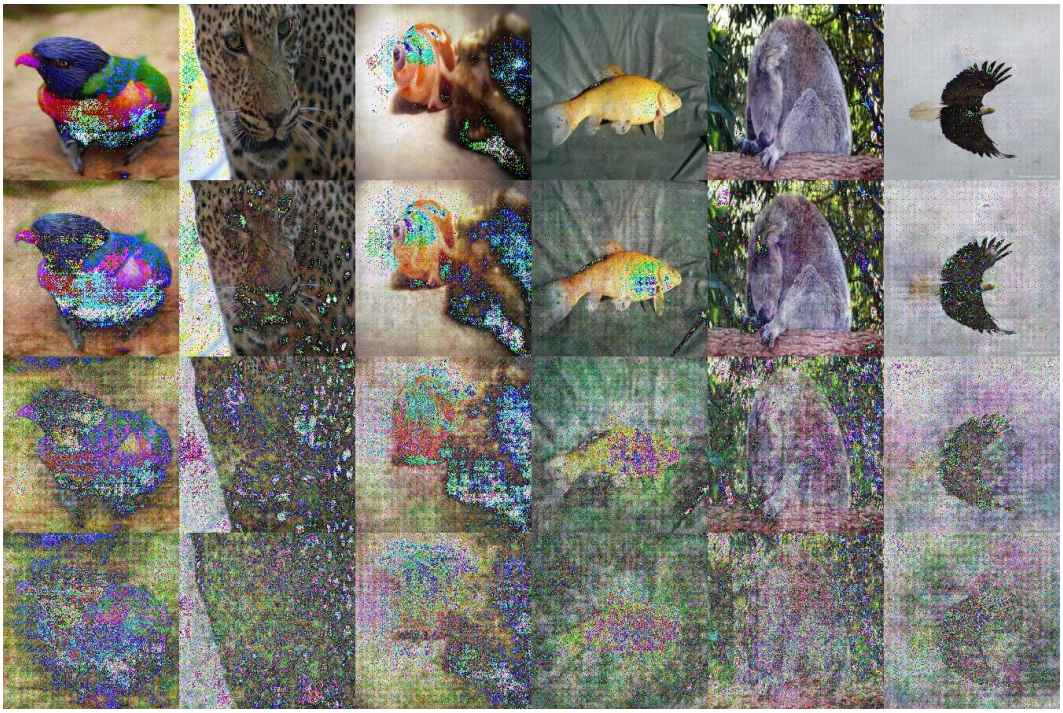

Figure 4: Third row: recovered images from features with 0.1% decay. Fourth row: recovered images from features with 1% decay. Fifth row: recovered images from features with 5% decay. Sixth row: recovered images from features with 10% decay.

5 Bridging Psychology

The purpose of this paper is not only to present an approach to computationally model human memory but also to imply a bidirectional relationship between artificial intelligence and psychology. Psychology, especially cognitive psychology, can inspire AI, and a computational model of cognition can be used to explain psychological concepts with computational logic.

5.1 Déjà Vu

Déjà vu refers to a common experience where a person has a weird feeling that the current experience has happened before in the memory, while as a matter of fact, it is impossible to have been through all this [3].

As mentioned in the previous section, the brain generates a feature for recalling. This is often unintentional. For instance, when we are travelling to a place which we never went to before. The brain will extract features from raw sensory information automatically. These features will be used for locating memory traces.

In most cases, as we never went to this place, the matched information unit has a significant difference in the feature space, so it will not be transferred into the working memory. However, in some scenarios, the matched information unit has a distance to the current explored feature less than the threshold. Then this memory trace will be recalled. This implies that the experience of déjà vu is related to the past memory, and there is indeed a memory trace that has a high similarity with regard to the features extracted from the current sensory information. In a nutshell, déjà vu happens when the brain executes unintentional recall with a mismatch on memory trace caused by low-quality features extracted.

5.2 Amnesia

Amnesia refers to partial or complete loss of memory. It can be temporary or permanent. Both physiological factors and psychological factors may cause amnesia [15]. Computationally, memory loss only has two possible causes: information incompleteness and search failure.

Information incompleteness is primarily caused by the decay of memory trace internally. If a memory trace does not get enough rehearsal, it will be decayed until it is removed from the memory. Even if a memory trace exists, it may be hard to recall because the decayed feature cannot pass the similarity check. The incompleteness may also be caused by external physical damage. For instance, if the hard disk for storing memory gets severe damage, the memory traces may be lost permanently.

Search failure has two cases. One case is the inability to locate the memory trace. Information decay may be the cause. Additionally, the cause can be that the features are of low quality, so the matching process cannot be executed as expected. The features mentioned above can be the feature for matching or the feature stored in the memory trace. Therefore, the low-quality feature extracted will cause search failure. Another case of search failure is that the status of the matched information unit is invalid. Usually, when a new information unit is created, the status is initialized to valid. However, if we implement the memory model to a robot with human-like emotions, the status of an information unit will be set to invalid if this memory trace causes a negative impact emotionally, like a traumatic experience. This is a protection mechanism for the emotional stabilization of an intelligent agent. The protection mechanism will protect the agent from recalling the traumatic experience. Additionally, the connected information units will not be recalled since the parent information unit has a status of invalid, so it cannot be transferred into working memory. Thus, there will be a chain effect of memory loss to multiple traces.

5.3 Curiosity

Curiosity is the impulse to explore information, especially when the material is novel [15]. With the model of memory presented in this paper, we can model curiosity computationally. When the brain is sensing, it executes unintentional memory searches. If the current generated feature has a big difference from the matched memory trace, it can be considered as a novel information unit. Thus, the attention of this information unit will add a positive floating-point number to simulate curiosity. This can be used to explain why humans remember novel information relatively longer.

6 Conclusions and Future Work

In this work, I present an approach to model human memory using invertible neural networks. It is a computational cognitive model (CCM) which is psychologically explainable, and it further demonstrates how CCM affects psychology.

To achieve more realistic modeling of human memory, this model can be extended by the followings:

1. Implementing another memory model: this paper uses the Atkinson-Shiffrin memory model (multi-store model) as the base model. However, other psychological memory models may be closer to actual human memory.
2. Processing more types of information: only visual information is demonstrated for encoding, storing and recalling in the memory, but it is better to have support for processing other sensory information. For example, NLP models can extract features from human languages. One important constraint is that the model must be invertible.
3. A better neural network model to extract features: deep learning models are improving all the time. A better model is larger, deeper with more parameters. Also, the neural network input should be larger than 224x224 because downsampling the image to this size will have a lot of information loss. More importantly, it should be able to correlate different sensory information (visual, auditory, haptic, olfactory and gustatory) because there are explicit and implicit relationships between them, which will make the model more "intelligent" if it can capture them.
4. A computationally efficient mechanism to simulate information decay: the decay simulation proposed in this paper is not low-cost enough. Therefore a better mechanism should be used in the decay simulation since we do not want to waste too many computing resources here.

References

[1] ARDIZZONE, Lynton ; KRUSE, Jakob ; WIRKERT, Sebastian ; RAHNER, Daniel ; PELLEGRINI, Eric W. ; KLESSEN, Ralf S. ; MAIER-HEIN, Lena ; ROTHER, Carsten ; KÖTHE, Ullrich: Analyzing inverse problems with invertible neural networks. In: *arXiv preprint arXiv:1808.04730* (2018)

[2] ATKINSON, Richard C. ; SHIFFRIN, Richard M.: Human memory: A proposed system and its control processes. In: *Psychology of learning and motivation* Bd. 2. Elsevier, 1968, S. 89–195

[3] BROWN, Alan S.: A review of the déjà vu experience. In: *Psychological bulletin* 129 (2003), Nr. 3, S. 394

[4] CLARK, Roger N.: *Clarkvision photography - resolution of the human eye*. – URL `https://clarkvision.com/imagedetail/eye-resolution.html`

[5] COWAN, Nelson: The magical number 4 in short-term memory: A reconsideration of mental storage capacity. In: *Behavioral and brain sciences* 24 (2001), Nr. 1, S. 87–114

[6] DENG, J. ; DONG, W. ; SOCHER, R. ; LI, L.-J. ; LI, K. ; FEI-FEI, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: *CVPR09*, 2009

[7] DEVLIN, Jacob ; CHANG, Ming-Wei ; LEE, Kenton ; TOUTANOVA, Kristina: Bert: Pre-training of deep bidirectional transformers for language understanding. In: *arXiv preprint arXiv:1810.04805* (2018)

[8] DOSOVITSKIY, Alexey ; BEYER, Lucas ; KOLESNIKOV, Alexander ; WEISSENBORN, Dirk ; ZHAI, Xiaohua ; UNTERTHINER, Thomas ; DEHGHANI, Mostafa ; MINDERER, Matthias ; HEIGOLD, Georg ; GELLY, Sylvain u. a.: An image is worth 16x16 words: Transformers for image recognition at scale. In: *arXiv preprint arXiv:2010.11929* (2020)

[9] GOLDSTEIN, E B.: *Cognitive psychology: Connecting mind, research and everyday experience*. Cengage Learning, 2014

[10] HE, Kaiming ; CHEN, Xinlei ; XIE, Saining ; LI, Yanghao ; DOLLÁR, Piotr ; GIRSHICK, Ross: Masked autoencoders are scalable vision learners. In: *arXiv preprint arXiv:2111.06377* (2021)

[11] HE, Kaiming ; ZHANG, Xiangyu ; REN, Shaoqing ; SUN, Jian: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, S. 770–778

[12] JACOBSEN, Jörn-Henrik ; SMEULDERS, Arnold ; OYALLON, Edouard: i-revnet: Deep invertible networks. In: *arXiv preprint arXiv:1802.07088* (2018)

[13] MILLER, George A.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. In: *Psychological review* 63 (1956), Nr. 2, S. 81

[14] SANCHEZ-LENGELING, Benjamin ; REIF, Emily ; PEARCE, Adam ; WILTSCHKO, Alexander B.: A Gentle Introduction to Graph Neural Networks. In: *Distill* (2021). – https://distill.pub/2021/gnn-intro

[15] VANDENBOS, Gary R.: *APA dictionary of psychology*. American Psychological Association, 2007

[16] VASWANI, Ashish ; SHAZEER, Noam ; PARMAR, Niki ; USZKOREIT, Jakob ; JONES, Llion ; GOMEZ, Aidan N. ; KAISER, Łukasz ; POLOSUKHIN, Illia: Attention is all you need, 2017