On Deep Research Problems in Deep Learning

Anwaar Ulhaq, *Member, IEEE*,
Machine Vision and Digital
Health Research Group,
Charles Sturt University, NSW,
Australia
aulhaq@csu.edu.au

***Abstract*—The subject of deep learning has emerged in the last decade as one of the most promising approaches to machine learning. Today, certainly, much of the recent progress in artificial intelligence is due to it, but research challenges are still unresolved and remain open to the research community. This paper attempts to offer a comprehensive review of deep learning progress in active research frontiers. On the one side, by presenting a brief overview of deep learning success, we inspire researchers to work in deep learning. On the other hand, we examine a range of technical issues, and open research issues that we believe are relevant topics for exploratory research. As deep learning applies to various fields, we restrict this paper's scope to visual recognition tasks to analyse these problems with a specific lens. However, these problems will be broadly applicable to other fields. It will make it easier for new researchers to recognise outstanding research problems in the deep learning domain.**

***Index Terms*—Deep Learning, Artificial Intelligence, Self-supervised learning, Contrastive Learning, Meta-Learning, Aderserial Learning.**

I. Introduction

Deep learning refers to a broad range of machine learning techniques that learn from large volumes of data. Deep learning algorithms are basically artificial neural networks that learn from data repeatedly, fine-tuning the task a little more each time. We call neural networks "deep learning" because they have many layers that allow for complex learning. The term "deep learning" refers to the number of layers from which data is processed [1].

A problem can be described scientifically as a general issue, concern, or controversy addressed in research. Additionally, to successfully conduct research, a problem must bring together existing concepts and theoretical perspectives to solve it. A research problem does not pose a vague or open-ended proposition, nor does it address a value issue. Therefore, researchers or students should focus on the problems to be solved, and the conditions to be improved, or the challenges to be overcome.

"I keep six honest serving-men (They taught me all I knew); Their names are What and Why and When And How and Where and Who" [2].

Many disciplines have used "honest serving-men" as the foundation for an investigation. However, this verse has been the most commonly quoted axiom of journalists since time immemorial, and it applies equally well to scientific research [3].

My ultimate goal for writing this paper is to help the reader understand the criticality of deep learning research so that the reader is prepared for what is about to happen and what will be discovered in the subsequent research questions or hypotheses. I want to define the parameters of what is to be investigated in deep learning for the long run. I have defined the following problems in deep learning after an extensive literature survey.

1) Performance (Accuracy vs Complexity Trade-off)
2) Scalability (Distributed Deep Learning, Deep learning on Cloud)
3) Optimisation (Parameter and Hyperparameter Optimisation, AutoML, Neural Architecture Search)
4) Generalisation (Regularisation, Domain Adaptation and Meta-Learning)
5) Data-Efficient Learning (Data Augmentation and self-supervised learning),
6) Interpretability and Explainability (Feature Visualisation and Explainable AI)
7) Security (Robustness, Safety and Reliability)
8) Fairness and Ethics (Privacy, Accountability, Transparency, Federated Learning and, Privacy by design)
9) Artificial Creativity (Genarative Models and Reinforcement Learning)
10) Artificial General Intelligence (Emotional intelligence, life loss)

This review paper is organised as follows: In section 2, we provide a list of top ten active research problem areas in deep learning . Section 3 presents the discussion and future research directions followed by a conclusion and references.

II. Active Research Problems in Deep Learning:

In this section, I will provide the detailed description of earch problem,with salient work and unanswered questins.

1) Performance:: The performance of deep models is a relative term as it depends on its design objectives and goals. Different deep neural network models (DNNs) are task-specific, like image classification, object identification, translation, speech-to-text, recommendation, sentiment analysis and reinforcement learning. Therefore, comparing their performance across different tasks would be irrelevant. Simultaneously, the difficulty of the objective task when comparing machine learning methods is also crucial. Difficult tasks typically require larger models. For instance, classifying MNIST handwritten digits [4] is much simpler than classifying objects into one of a thousand classes for the ImageNet dataset [5].

TABLE I
TOP PERFORMING DNN MODELS ON IMAGENET [6].

Rank	Model	Top 1 Accuracy	Top 3 Accuracy	Number of Params	Extra Training Data	Paper	Code	Result	Year
1	Meta Pseudo Labels (EfficientNet L2)	90.2%	98.8%	480M		Meta Pseudo Labels	yes		2021
2	Meta Pseudo Labels (Efficienet B6-Wide)	90%	98.7%	390M		Meta Pseudo Labels	yes		2021
3	NFNET-F4+	89.2%		527M		High-Performance Large Scale Image Recognition Without Normalization	yes		2021
4	ALIGN Effiecient	88-64%	98.67%	480M		Scaling Up Visual and Vision Language Representation Learning With Noisy Text Supervision			2021
5	EfficientNet L2-475 (SAM)	88.61%		480M		Sharpness Aware Minimization for Effieicently Improving Generalization	yes		2020

A software and hardware design specifications are important to know whether the design's objective is to create a DNN model that is accurate with efficient hardware or accurate for software-based implementation. DNNs are powerful tools for providing state-of-the-art accuracy on many AI tasks but at the cost of high computational complexity. Accordingly, designing efficient hardware architectures for deep neural networks is an important step towards enabling the wide deployment of DNNs in AI systems. It requires consideration of a comprehensive set of metrics when comparing DNN performance.

Design tradeoffs should be evaluated in an equitable manner across all performance metrics. For a given task, a broad suite of DNN models can be used as a common set of benchmarks to measure the performance and enable fair comparison of various software frameworks, hardware accelerators, and cloud platforms for both training and inference of DNNs. Another important tradeoff to compare the performance goals of DNS is the understanding of accuracy vs complexity tradeoff. Therefore, DNN models can be grouped into two categories [7] – High Accuracy DNN Models: Designed to maximise accuracy to compete in the ImageNet Challenge – Efficient DNN Models: Designed to reduce the number of weights and operations (specifically MACs) while maintaining accuracy. Overall, learned models have improved accuracy vs. 'complexity' tradeoff compared to handcrafted models.

Classification models can be judged on their degree of accuracy. Formally, accuracy is the percentage of our predictions that were correct. Formally, accuracy is the number of correct predictions divided by a total number of predictions. The accuracy of the model gives us information about its quality. While you only want to measure the accuracy, you must look at all aspects of your results, especially if you are working with a class-imbalanced data set in a situation where there is a significant number of positive and negative labels. Loss function also quantifies the agreement between the predicted scores and the ground truth labels. In other words,– Class that matches the ground truth label has the highest score – Classes that don't match the ground truth label has low scores. Quantifying loss allows us to improve the classifier (i.e. update weights) –how good is the classifier? Precision, recall, AUC, ROC curve, mAP are other metrics [9]. Overall, these metrics measure the performance of DNN in terms of the quality or state of being correct or precise.

Researchers on hand are trying to find out those factors that affect the accuracy of DNN. For instance, One early observation about deep learning was the dependence on accuracy on the size of data. The accuracy of the deep learning algorithm should be measured on a sufficiently large dataset as accuracy increases logarithmically based on the amount of training data.

On the other hand, most DNN researchers use the number of weights and operations to measure the "complexity" of the model. A measure of complexity is throughput. The throughput is dependent on both the amount of computation and the dimensionality of the data. It becomes more important for measuring the performance of real-time deep learning systems. The number of weights indicates storage cost for inference design tradeoffs.

Similarly, for interactive applications (e.g., autonomous navigation), latency is important. Increasing throughput and reducing latency is a key design objective of DNN. Other hardware related metrics are Energy and Power, flexibility (Range of DNN models and tasks) and scalability (Scaling of performance with the number of resources). Higher dimensionality produces more data, and programmability means that the weights must be read and stored as well. Because of the expense of data movement, energy efficiency is difficult to maintain. Storage requirements drive the cost. A systematic way of identifying performance limits for DNN hardware is a function of the DNN model and hardware design characteristics. Different layer shapes impact the amount of required storage and compute where available data reuse can be exploited. Even though the number of operations doesn't directly translate to throughput – The number of weights and operations doesn't directly translate to power/energy consumption. The understanding of the underlying hardware is important for evaluating the impact of these "efficient" CNN models. However, for the sake of simplicity, we may consider complexity as inefficiency in computation by both software and hardware.

An automated process for evaluating whether a DNN solution is a viable option for a given application might go as follows: accuracy is what decides whether or not it can complete the given task Latency and throughput determine whether it can meet performance and responsiveness requirements. Since

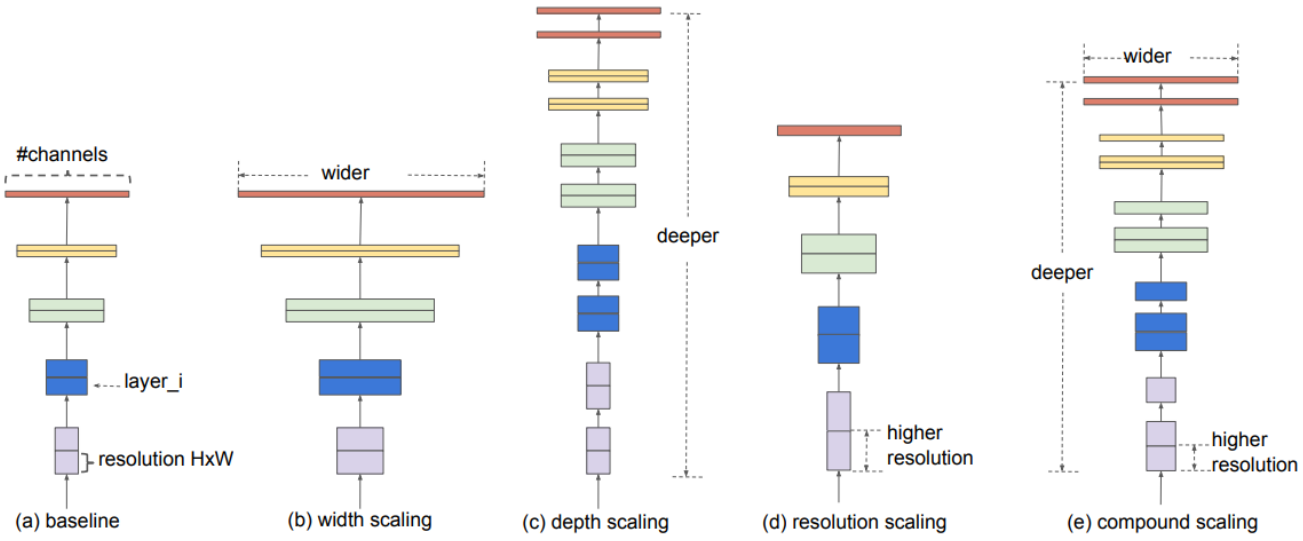

Fig. 1. Various scaling models for improve performance [8]

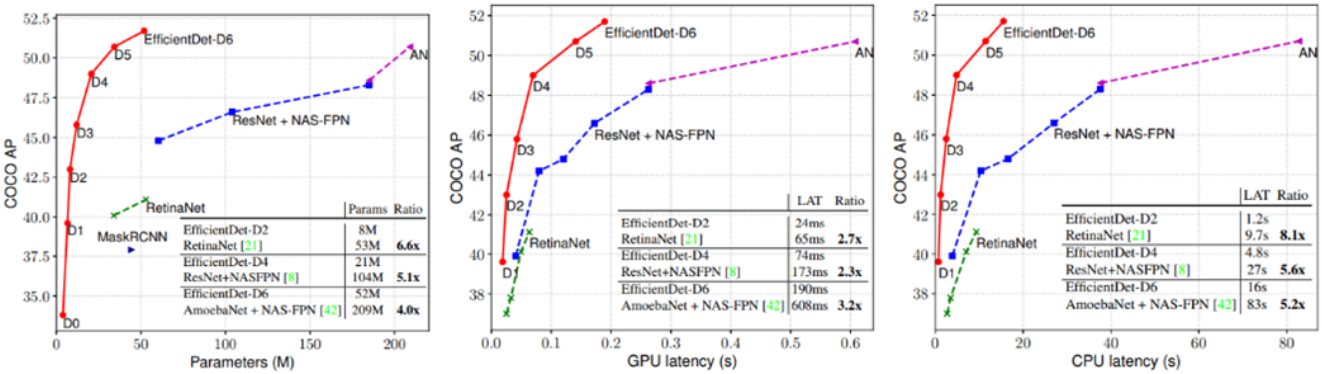

Fig. 2. Top performing algorithms on Imagenet in terms of Top-1 accuracy [6]

the processing can only run in one device, its physical form factor will be based on energy and power consumption. The primary cost driver is the chip area and the external interfaces. Determines how much one would pay for this solution 5. Flexibility determines the range of tasks it can support [7].

This discussion motivates us to ask the following research questions:

1- What factors affect the accuracy of DNN models irrespective of the underlying hardware efficiency? 2- What factors affect the efficiency of DNN models without sacrificing their accuracy? 3- How can we deep learning models achieve better performance to human in all types of learning?

TABLES Most accuracte models.Most efficient Models, tradeoff

2) Scalability: Deep learning is likely to be used everywhere. Adapting to the system's constraints will likely lead to inaccuracy. Engineers will, rather, have to design systems that can dynamically vary the type of processing resource they offer based on the task at hand. As long as big data and the cloud continue to flourish, compression and scalability will be more important than ever.

Distributed deep learning systems (DDLS) [10] use a cluster's distributed resources to train deep neural network models. A DDLS is required to make numerous decisions to process their specific workloads efficiently in their chosen environment. As GPU-based deep learning, larger datasets, and deep neural network models continue to grow, along with bandwidth constraints in cluster environments, developers of DDLS need to remain nimble to train high-quality models quickly. It is hard to compare DDLS side-by-side because they have completely different feature lists and architectural differences [11].

To deploy deep learning algorithms into production, it is essential to apply deep model efficiently. For deep learning and data analytics to be combined in a unified workflow, new projects like BigDL [12] (a distributed deep learning framework for Big data TensorFlow platforms and Workflow and implemented as a library on top of Apache Spark [13]. BigDL directly uses functional models to implement distributed and parallel training on top of copy-on-write and coarse-grained operations It provides an opportunity for Production-ready DL (a distributed deep learning framework used in various pro-

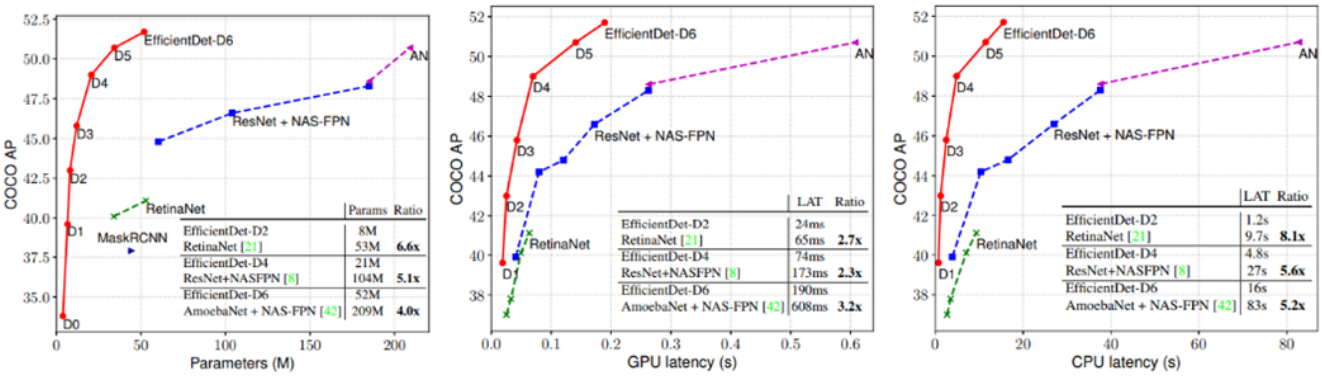

Fig. 3. Model size and inference latency comparison. Source [8].

duction environments for building deep-based applications). Hadoop Spark/Hadoop allows deep applications to be run on the Apache Hadoop cluster to directly process the production data while also being integrated into the end-to-end analysis pipeline [14].

Semiconductor manufacturers are hard at work in developing entirely new architectures for executing deep learning models more efficiently. There is no doubt the development of GPU has contributed to the development of deep learning. Next in processor evolution comes around. NPU is a name for the new processor architecture that will make machine learning available to a wider range of applications [15]. Future-proofing of the hardware platforms against new software applications is facilitated by choosing scalable architectures composed of MCUs, CPUs, GPUs, and NPUs [16]. Incremental learning and online learning necessitate low-cost, high-capacity neural networks with many layers and nodes. Scalable Deep Learning services are dependent on a number of factors [17]. The target application determines whether it requires low latency, enhanced security, or long-term cost-effectiveness.

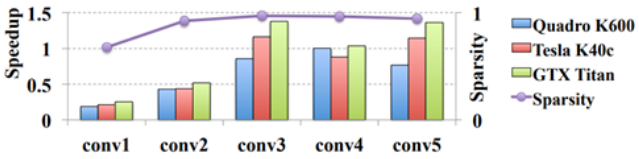

Fig. 4. Evaluation speedups of AlexNet on GPU platforms and the sparsity. conv1 refers to convolutional layer 1, and so forth.Source [18].

At the same time, deep learning is becoming more accessible due to cloud and edge computing services [19], [20] [21]. Cloud makes it easier to manage large datasets and train algorithms on distributed hardware. Distributing model training across multiple machines is made possible through the provision of large-scale computing capacity on demand. Special hardware configurations, such as GPUs, FPGAs, and massively parallel high-performance computing (HPC) systems, are made available to the deep learning community over the cloud. In addition, complex features like handling datasets and algorithms, training models, and deploying them to production are provided by cloud services. Running deep learning workloads in a "do it yourself" model is possible in each of these clouds. It involves selecting machine images that are already installed with deep learning infrastructure and running them in an IaaS (such as Amazon EC2 instances or Google Compute Engine VMs) model.

One critical feature of a cloud-based deep learning service is seamlessly integrating with notebooks and sending training jobs to cloud-based compute instances [22]. Google's portfolio of machine learning services is collectively known as Cloud AI [23]. The portfolio includes general-purpose and dedicated services for different use cases. AWS SageMaker [24] provides the Amazon Web Services service, which gives you the ability to build and manage machine learning models on the cloud, with a particular emphasis on deep learning. In other words, Azure Machine Learning [25] is a full machine learning ecosystem: training, deploying, and managing models.

However, choosing the cloud as a place to host your deep learning model isn't necessarily the best solution in some scenarios like their usage for the internet of things (IoT) devices. Edge-based deep learning offers additional benefits. Here, the term "edge" means a local computation performed on the consumer's products. For instance, commercially-oriented servers are susceptible to attacks and hacks. Better control over IP is possible if devices are on edge. However, it is widely known that deep learning models are large and computationally expensive. Edge devices usually have cheap memory, so it's a challenge to fit these models into all kinds of systems. In general, edge devices are unable to handle large neural networks. Training DL models on edge devices are still difficult even if we use pre-trained DL models. It motivated researchers to maintain accuracy while limiting the size of the neural networks. Various possible architectures for fast inference on IoT devices is discussed by [21] that includes: 1) on-device computation, where DNNs are executed on the end device; 2) edge server-based architectures, where data from the end devices are sent to one or more edge servers for computation; and 3) joint computation among end devices, edge servers, and the cloud.

On similar footings, virtual reality, augmented reality, and smart wearable devices [26] provide tremendous opportunities for researchers to pursue complex deep learning challenges on limited-resource portable devices (e.g. memory, CPU, energy,

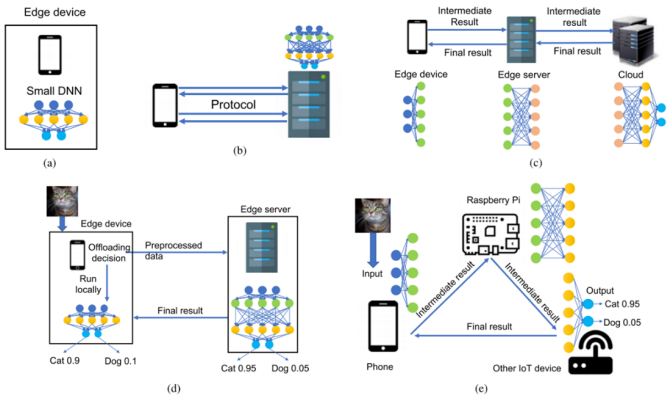

Fig. 5. Different Architectures visualised by [21] for deep learning inference with edge computing. (a) On-device computation. (b) Secure two-party communication. (c) Computing across edge devices with DNN model partitioning. (d) Offloading with model selection. (e) Distributed computing with DNN model partitioning. Source [21].

bandwidth). Distributed systems, embedded devices, and FPGAs can all benefit from efficient deep learning methods. For example, the ResNet-50 [27] requires 95MB of memory and 3.8 billion floating-point multiplications to process an image. Redundant weights are discarded, and the network functions as usual, but it saves 75% of parameters and 50% of computation time. Specifically, 2D-Torus all-reduce arranges GPUs in a logical 2D grid and performs a series of collective operation in different orientations. It can reduce its computation time by successfully trained ImageNet/ResNet-50 in 122 seconds without significant accuracy loss on a cluster [28]. However, with small resource, how can we compact the models?

Deep model compression [29], [30], therefore, becomes a crucial area of research if we want to deploy deep learning systems on smart devices. The removed parameters and layers reduce the complexity, which results in the application being well-suited for the edge. Optimal DL Implementation [19], [31], [32], [33] optimises the DL models to be more suitable for the edge. These implementation use Approximation (to use uncertainty estimation to make DL more suitable for the edge devices.), Model Parallelism (to distribute the memory and computational requirements by distributing the model itself) and Data parallelism (distributing data between clients and collaboratively learn a shared model). SOAP [33] goes beyond model and data by including strategies to parallelise a DNN in the Sample, Operation, Attribute, and Parameter dimensions.

Some model compression approaches are valuable in this regard: Network pruning [34], [35] can remove unnecessary, less relevant, or sensitive links from the network, memory to reduce the demand for computation power and to increase the speed. The quantisation techniques [36], [37] can change the presentation of each parameter, i.e. 32-bit float to 8-bit or less. Hashing [38], [39] can provide low-cost hash functions to randomly cluster connection weights into hash buckets that share the same parameter values. The parameter pruning, hashing and quantisation methods explore model parameter redundancy and seek to remove redundant and uncritical parameters. Low-ranking factorisation based techniques [40], [41] employ matrix/tensor decomposition to estimate informative parameters of the DNNs. Convolutional filters based on methods utilise compact/transferred convolutional techniques design specially-tailored convolutional filters to reduce the parameter space and, to save storage and computation. It includes weight sharing to simplify network architecture by mixing weights between layers or structures (e.g filters in CNNs) [42], [43]. A similar technique is structural sparsity learning that can reduce a ResNet architecture with 20 layers to 18 with a 1.35 percentage point accuracy increase on CIFAR-10 [18]. Knowledge Distillation [44], [45] helps to learn a small network from a large network by incorporating supervision from the larger network and removing the amount of entropy or variance. These methods identify a distilled model and then train a smaller neural network to reproduce the output of a larger one.

Overall, the challenge of scalability is maintaining the accuracy of the system by introducing efficiency. It can be achieved either by hardware architecture that supports scalability and algorithmic architecture that supports compression. It raises few questions:

1-What factor impact the design of Scalable Deep Learning services for cloud and big data? 2-What innovation do we need to bring to semiconductor architecture design to support scalable deep learning? 3-How can we simplify the training and testing of DL models on edge devices and IoTs with scalability and elasticity? 4-How can we compact deep learning models on cell phones and FPGAs with small and a few resources 5-How can we begin training with a smaller network, self-contained network and obtain the same or better generalisation performance?

TABLES: Cloud Models,Network Compresson

3) Optimisation: Parameter and Hyperparameter Optimisation: In deep learning, we formulate a task as a problem and train a neural network to solve it. The model includes architecture and specifications. An architecture's parameters control how effectively the model performs the task. A model parameter is an internally defined component whose value can be calculated from observations, examples of which include weight parameters. It also includes hyperparameters that cannot be estimated from data. A hyperparameter must be manually specified if it is critical such as the learning rate. Modelling parameters are typically discovered using an optimisation algorithm, which is an efficient exploration of possible values. Therefore, applying optimisation to mathematical models is an important part of deep learning. The availability of highly effective optimisation is essential to all deep learning processes. On the other hand, new computation methods, such as deep learning, provide new tendencies and new insights for optimisation. Choosing the right optimisation algorithm will make a huge difference in the accuracy.

The model parameters are iterated until the desired prediction is obtained: We run the data through the model's operations, compute the accuracy, and tweak the values until we find the optimum. Often, the model defines a loss function that measures how well it performs. The goal is to minimise loss and then identify parameters that correspond to reality. The essence of training is knowing what to do when the going

gets tough. In neural networks, training is typically iterative and time-consuming. It is in our interests to reduce the training time as much as possible. This term is known as "parameter optimisation," aiming to select the most effective parameter values (e.g. minimise an objective function you choose over a dataset you choose).

Gradient Descent Algorithm [46] provides the Treasure-map for deep learning in deep learning as it helps hunt the treasure (minima of a cost function) successfully. Imagine a group of blindfolded explorers (data points) that are sent to a treasure-hunting mission (finding the global minima of the cost function) in a steep mountainous valley (possible values of loss function). Their mission is to find the lowest point in this valley (global minima). They also need to find hidden values of important parameter, e.g. weight of their bag pack (weights) and value of basic gear (bias), so that can lead anyone to the treasure location. They are blindfolded, so they cannot see but are provided a gadget that only provides one important clue: the slope value (the gradient of the cost function). There are some constraints like only one step (update) can be taken at a time. Step size value (the learning rate) is the choice of explorers. In this scenario, the gradient descent would be the mission plan.

Let us add some important heuristics: As moving in the direction of the gradient (increased height) is not helpful to find the lowest place in the valley, moving in the direction of negative of the gradient would be helpful. So, initialise the parameters randomly, calculate the loss over all the dataset at once, calculate its gradient and take the step in the opposite side of the gradient. Once we complete one complete cycle, we call it an epoch. It is called Vanilla Gradient Descent. However, as we calculate the over all loss dataset, it is a very slow update proportional to the number of data points. What if we replace the whole dataset's gradient with an estimate, the gradient of a randomly selected data point at each step to calculate the gradient. The sample is randomly shuffled and selected for performing the iteration, and we can save a lot of time at the cost of some noise, and it is called Stochastic Gradient Descent [47]. How about sampling a small group (Mini-batch) of data points than the complete dataset? It is called Mini-batch Gradient Descent [48].

There are still two problems: explorers can stick into local pit or minima, and they may keep on wandering or wasting time in the plateaus or flat areas (areas with low gradient). We can solve this problem through another clue call momentum. The idea is if their gadget is constantly or repeatedly pointing in the same direction, they should develop some confidence in that direction and start running than walking. This confidence or momentum is gained through a history gradient (exponential decaying cumulative average of previous gradients). So, the explorer can more use two indicators, the current gradient and the history gradient (momentum). It can help them move faster, especially in flat surface areas and called Momentum-based Gradient Descent [49]. However, moving faster is not always good as in potential areas where minima are present, it can show oscillation behaviour and get past the minima point due to high momentum resulting in missing the minima for a while until it decides a U-tun. Nesterov provided a better plan: Rather than taking one move according to the current gradient and the other based on history gradient, how about moving a little in the direction of history gradient, then calculating the current gradient and then update the route parameters? in other words, before proceeding, we're hoping to see if we're closer to the minimums or not and it is known as the Nesterov's Accelerated Gradient-Descent [50].

However, we are assuming that all explorers step forward in equal amount or their step size (learning rate) is the same as usual. It also depends on the surface or trail history. Suppose they are constantly walking down the steep track (derivative is non zero most of the time, frequent updates, dense variable like b) for the sake of safety. In that case, they are required to reduce their step size (learning rate). Those whose track history is mostly in flat surfaces (with zero derivatives most of the time, fewer updates, sparse variable like w) are alright to move ahead with a large effective step size (learning rate). It can be achieved by dividing the learning with the history of gradient value. This variant of gradient descent is called AdaGrad [51] with the idea of adaptive learning rate. However, there is another problem as the decreasing learning rate becomes closer to zero, AdaGrad can get stuck closer to the convergence in the case of parameters with frequent updates. Can we prevent this rapid decay of dense variables? This solution was proposed by RMSProp (Root Mean Square Propagation) [52], another variant of gradient descent. It changed the calculation of the history of gradients. In AdaGrad, it was being calculated by a simple sum of gradients; it changed it by exponential decaying average function. It introduces control of rapid decay of dense variables. It is guaranteed to lead to convergence (treasure hunt).

The last variant is called Adam (Adaptive moment estimation) [53]. All we have learnt so far is the history is such an important cue. In momentum-based DG, we used the history of gradients to speed up and move faster in flat areas; in RMSProp, we used it to control the learning rate (the step size). Both histories are important, so Adam uses both to devise a faster and greenfeed treasure hunt plan. It is considered as one of the best plans but has a legacy of momentum, so explorers will need to have few U-turns before they find the treasure.

A hyperparameter is a parameter whose value is set before the learning process begins. Hyperparameters tuning is crucial as they control the overall behaviour of a deep learning model. Every deep learning model will have different hyperparameters that can be set. Hyperparameters are tuned by running the whole training job, looking at the aggregate accuracy, and adjusting. The optimal settings for hyperparameters are often different for different data sets. Tuning is required to find the best settings for each dataset. The training process itself does not determine the hyperparameters; therefore, a meta-process is required to fine-tune the hyperparameters [54].

Hyperparameter tuning is what we refer to as Hyperparameter -optimisation (or tuning) [55]. It refers to a procedure that selects the best parameters for a deep learning algorithm. Everything in the data-processing pipeline (including data preprocessors, optimisers, and algorithms) gets parameters to guide their behaviour. To achieve the best results, values need

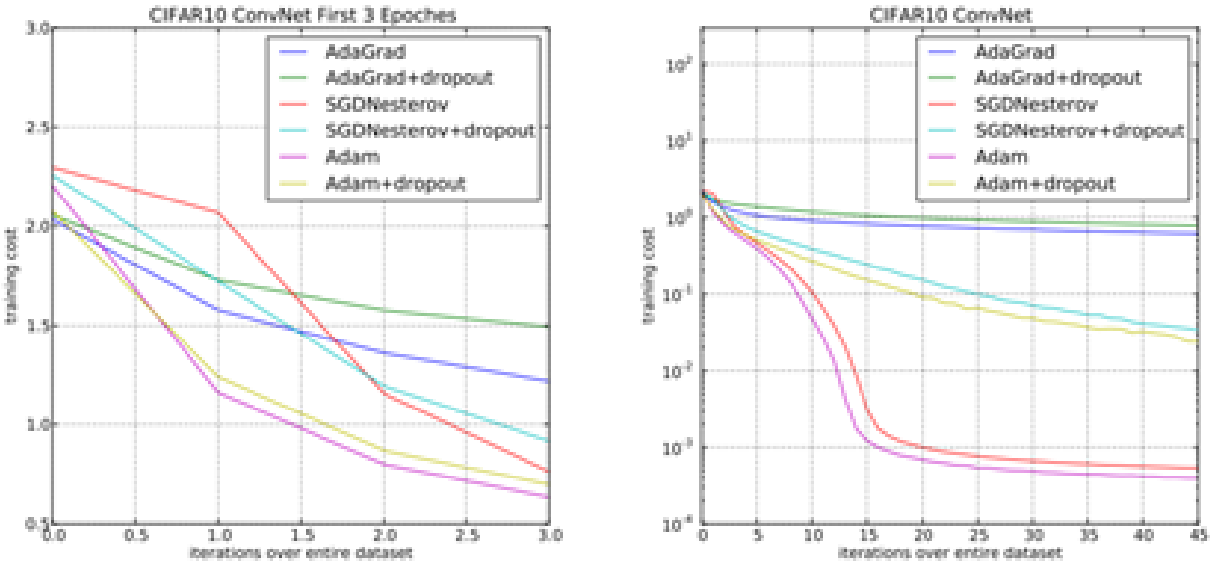

Fig. 6. Convolutional neural networks training cost. (left) Training cost for the first three epochs. (right) Training cost over 45 epochs. CIFAR-10 with c64-c64-c128-1000 architecture. Source [53]

to be tuned to the characteristics of the data, type of features, and size of the dataset. The most typical hyperparameters in deep learning settings include learning rate, neural network layer count, and dropout Search grow exponentially with the more parameters we tune. However, the big question is: which hyperparameters should be selected for maximum model performance?

The most likely to include the first step of hyperparameter optimisation is to include defining the search space [55]. This means that we must provide all the hyperparameters we want to tune and verify and specify whether the optimiser can search in a discrete space or a continuous one. There are various optimisation techniques such as grid search, random search, the Parzen tree approach, and Bayesian optimisation (TPE) [55] [56].

To tune a parameter is to find the best possible hyper-parameter values, and it is an optimisation like training a model. While these two objectives may be similar in theory, they are very different in practice. When building a model, the parameters' quality may be defined as a mathematical equation (usually called the loss function). When trying to tune hyperparameters, there is no possibility of writing them in a closed-form because it depends on the outcomes of a black box (the model training process). This is why tuning machine learning hyperparameters is comparatively difficult. Until recently, only the grid search and random search methods were available. The term "Grid search" means to perform a search and return the winner within a hyperparameter grid. In terms of computation time, it is the most expensive option. Conversely but, if performed serially, it runs fast with respect to time. The random search function is a change in the conventional search. Rather than searching the entire grid, search a random sample of locations only. This saves a lot of money when searching randomly. A random search was considered insignificant. This is because it does not search across all points, so it cannot achieve the best result in the grid search. As opposed to brute-force and random methods, hyperparameter tuning is less parallelisable.

Automated machine learning [57] refers to the discovery of well-performing models without much user involvement. Some frameworks include the Google Cloud AutoML [58] or open-source solutions such as the NNI (Neural Network Intelligence) on Github [59]. Using AutoML can help select the appropriate hyperparameters for the model.

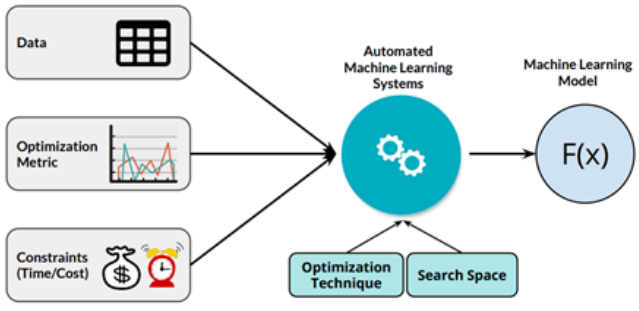

Fig. 7. A generic workflow of AutoML processes. Source [60]

While human experts usually drive the popular and successful models, it doesn't mean we have explored the entire network space and settled for the best option. The various subfield of automated machine learning that is most closely related to automated hyperparameter optimisation is Neural Architecture Search (NAS) [61]. Network engineering is made easier with Neural Architecture Search (NAS). Neural Architecture Search (NAS) [61], is used to automate the manual process of designing neural networks. It strives to

learn the most effective network topology for a certain task by breaking the methods down into three parts: search space, search algorithm, and child strategy evolution [61], [62].

Neural architectures can be searched or discovered using a variety of search strategies. Different search methods have been tried out over the past two or three years includes Neural Architecture Search with Reinforcement Learning (such as [63], NASNet [64] and ENAS [65] [66]. Evolutionary algorithm (such as Hierarchical Evo [67] and AmoebaNet [68], Sequential model-based optimisation [69], Bayesian optimisation (such as Auto-Keras [70], and Gradient-based optimisation (such as SNAS [71] and DARTS [72]. This research area is readily evolving.

In the last few years, interest in self-tuning has increased, and several research groups have conducted various studies, published papers, and new tools [57], [73], [74], [60]. However, there are many important research questions related to the parameter, hyperparameters optimisation and AutoML like:

How can we improve the optimisation of gradient descent methods beyond the ADAM approach? How can e find the winner or one-size-fits-all technique hyperparameter optimisation for AutoML techniques? How can we reduce the search space of AutoMLtechniques and make them scalable? How can we optimise the time budget for AutoML? How can we increase the reliability and validation of AutoML methods?

4) Generalisation: Generalisation is an important concept that human and animal learning applies both in similar current and new situations [75]. The knowledge transferability of the subject is directly connected to this capacity. When learning new information, a rule, people often refer to it as representations, as an abstraction, because they transfer representations that are comprised of characteristics similar to their prior knowledge. Artificial intelligence learns to identify different categories by drawing on prior knowledge when facing novel situations, using prior knowledge relevant in one or more ways.

Generalisation in deep learning represents how well a model can handle previously unknown data from the same distribution. The issue here is that if a network can correctly represent training data, will it also do so on test data? [76]. Generalisation is the difference between merely memorising information from training data and acquiring true knowledge about it. Two concepts are important: Underfitting and overfitting [77]. Underfitting means poor performance on the training data while poor generalisation to other data. Overfitting means good performance on the training data and poor generalisation. Here, we have a performance discrepancy between the model on training data, which is customary. The greater the overfitting in training data, the narrower the generalisation. However, we still struggle to explain large artificial neural networks' generalisation ability as it is a mystery to understand why some models generalise better than others. Model parameters and the optimiser cannot accurately explain state-of-the-the-art neural networks' effectiveness due to their variable effect on the results.

In machine learning, regularisation is believed to help in reducing overfitting [77]. Implicit regularisation refers to the learner's preference to implicitly choosing structured solutions as if some explicit regularisation term appeared in its objective. Zhang et al. [78] show how the properties of stochastic gradient descent) SGD) acts as a regulariser. For linear models, every SGD solution converges to a small norm. It is also shown that sharp minima lead to poorer generalisation. In contrast, small-batch methods consistently converge to flat minimisers [79].

Statistical theory suggests that when the number of parameters is large, some form of regularisation is needed to ensure a small generalisation error. However, model architecture doesn't contain sufficient regularisation (overfitting prevention)" the main regularisation techniques used are L1 and L2 [80]. These include the regularisation term, in addition to the cost function's main assumptions. This regularisation term assumes that smaller weight matrices result in simpler models, overfitting will therefore be reduced. Weight decay forces the weights to reduce towards zero (but not exactly zero).Therefore, it will also reduce overfitting reasonably. L2 regularisation is also considered as weight decay as it forces the weights to decay towards zero. In L1, the weights may be reduced to zero as we are trying to compress our model. Dropout [81] is one of the most interesting techniques for regularisation. It also produces excellent results and is widely used in the field of deep learning. This type of regularisation is also called explicit regularisation. Batch normalisation is usually found to improve generalisation performance. Zhang et al. [78] show that The normalisation operator helps stabilise the learning dynamics, but the generalisation performance's impact is only 34.

Data regularisation is another strategy as it provides reasonable overfitting control. It increases both the amount and diversity of data by randomly "augmenting" it [82]. AutoAugment is a way to automatically search for improved data augmentation policies [83], [84], where reinforcement learning is used to search for better data augmentation policies.

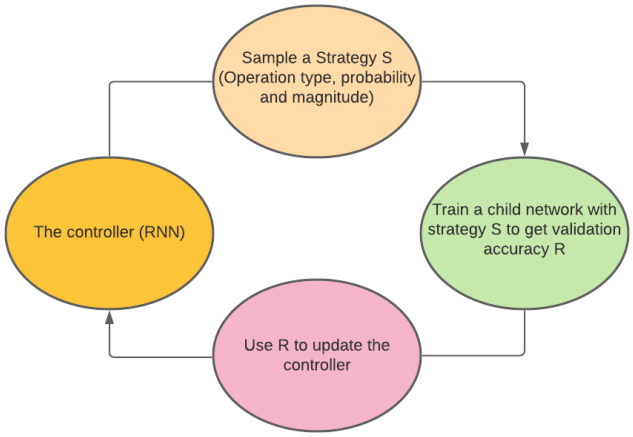

Fig. 8. A controller RNN predicts an augmentation policy from the search space. A child network with a fixed architecture is trained to convergence, achieving accuracy R. The reward R will be used with the policy gradient method to update the controller to generate better policies over time.

Zhang et al. [78] show that formal, explicit forms of regularisation such as weight decay; dropout doesn't adequately

explain the network generalisation. Explicit regularisation may improve generalisation but is neither sufficient nor necessary for doing so.

There is an interesting case of random labelling of data where the true labels are replaced by random labels.. Such an attempt to understand the generalisation was made by changing the network's different parameters by Zhang et al. [78]. Several experiments varied the degree and type of randomness across the dataset. It includes (i) true labels (original dataset without modification), (ii)partially corrupted labels (mess with some of the labels), (iii) random labels (mess with all of the labels), (iv) shuffled pixels (Using a particular pixel permutation to all of an image), (v) random pixels (use random permutations on every image) and (vi) Gaussian (applied on for each image) and collected results on CIFAR data set. It is surprising that along the spectrum, the networks are still able to fit the training data perfectly.

It turns out that by randomising labels alone, we can force the generalisation error of a model to jump up considerably without changing the model, its size, hyperparameters, or the optimiser. Even if images are replaced by completely random pixels, CNN (e.g., Gaussian noise)show a steady deterioration of the generalisation error as there is an increase in the noise level. Though it does not address why some models generalise better than others, this does provide evidence that more exploration is required to learn more about what is common to all models. It is extremely important to understand the generalisation gap and how it applies to the optimisation procedure.

Wider use of over-parametrisation has arisen in deep learning, as it is now widely observed larger neural nets can achieve better performance. Furthermore, a larger network can be trained to achieve a certain level of prediction performance with fewer iterations than a smaller net. This observation, to our knowledge, can be dated back as early as the work of Livni et al. [85]. They tried different over-parametrisation levels and reported that SGD converges much faster and finds a better solution when used to train a larger network. However, the reason why over-parametrisation can lead to an acceleration remains a mystery.

A lot of work is done on generalisation from an empirical point of view. Perhaps, one issue we must work on is: Is there a model or data-related measure we can use to calculate the model's general properties?

The most common (and traditional) tools to study the generalisation capabilities of a model are its VC dimension and its Rademacher complexity.In Vapnik–Chervonenkis theory [86], the Vapnik–Chervonenkis (VC) dimension is a measure of the capacity (complexity, expressive power, richness, or flexibility) of a set of functions that can be learned by a statistical binary classification algorithm. It is defined as the cardinality of the largest set of points that the algorithm can shatter. The capacity of a model is related to how complicated it can be. Recently, Morcos et al. [87] show that Area Under Cumulative Curve, or AUC, is a good generalisation prediction of network performance. The networks with higher AUC performed better. These tools can be used to establish upper bounds on the difference between training and testing performance.

Generalisation is understood to be directly tied to the transfer of knowledge across multiple situations. We humans have a built-in capability to switch information between tasks. Learning to ride a motorbike (of any kind) greatly facilitates getting around the city but does not come naturally to everyone master how to drive a car. Training neural networks from one task to other tasks is known as transfer learning [88], [89], [90]. Transfer learning allows you to use information (characteristics, quantities, key-values, weight values) from previously trained models to help train models for other purposes. Different approaches and techniques can be employed depending on the domain, mission, and data availability. Transfer learning has emerged as a strong discipline in the context of deep learning. According to Goodfellow [5], Transfer learning and domain adaptation refer to the situation where what has been learned in one setting ... is exploited to improve generalisation in another setting. Leveraging knowledge (features, weights etc.) from previously trained models can lead to improved performance, and generalisation as it enables us to utilise knowledge from previously learned tasks and apply them to newer, related ones.

The concepts of domain and task are fundamental to transfer learning. In Inductive Transfer learning, the source and goal domains are the same, but the target tasks are different. In Transductive Transfer Learning, there are similarities between the source and target tasks, but the corresponding domains are different. These can be further partitioned with respect to where the function spaces differ, or the probabilities differ.

Domain adaption [91] is a special case of transfer learning that usually referred to in scenarios where the marginal probabilities between the source and target domains are different. So in domain adaptation, our goal is to train a neural network on one dataset (source) for which label or annotation is available and secure good performance on another dataset (target) whose label or annotation is not available. These techniques are generally divided into three categories of (i) Divergence based Domain Adaptation [92], (ii) Adversarial based Domain Adaptation [93] ,(iii) Reconstruction based Domain Adaptation [94]. The first type of methods use minimising some divergence based criterion between source and target distribution, hence leading to domain invariant-features like Maximum Mean Discrepancy (MMD). The send category uses GANsAs Here; the generator is simply the feature extractor and discriminator networks learn to distinguish between source and target domain features. The third category uses the idea of Image-to-Image translation. One of the simple approaches could be to understand the translation from the target domain images to the source domain image and train a classifier on the source domain.

Overall, achieving better generalisation is one of the highest goals of deep learning and still various questions need answers:

What is it then that distinguishes neural networks that generalise well from those that don't? What factors can increase the generalisation performance of any deep learning algorithm?

5) Data: Data is the food of deep learning, and the availability of big data is an enabler of the deep learning revolution. In one issue of the Economist, it published an article titled,

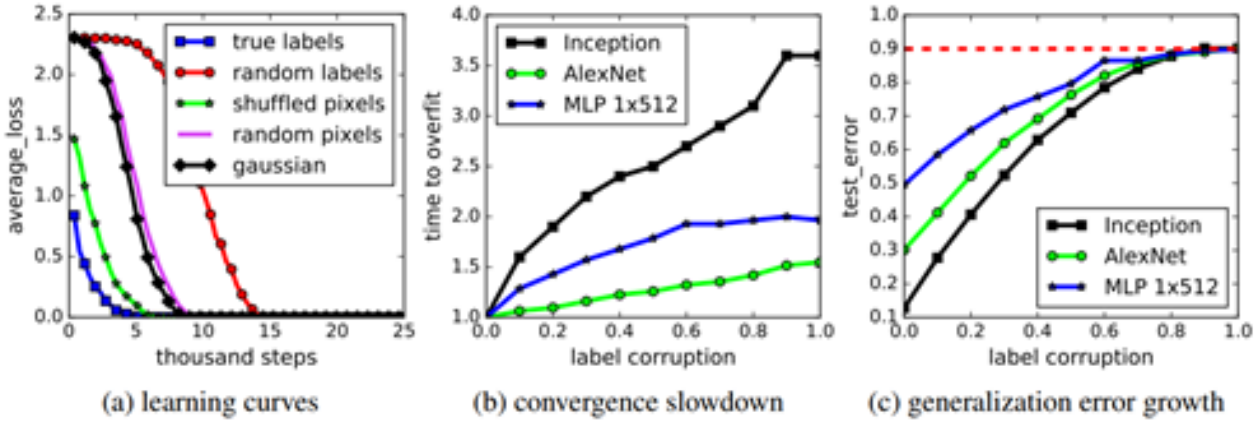

Fig. 9. The curious case of random labels. Source [78]

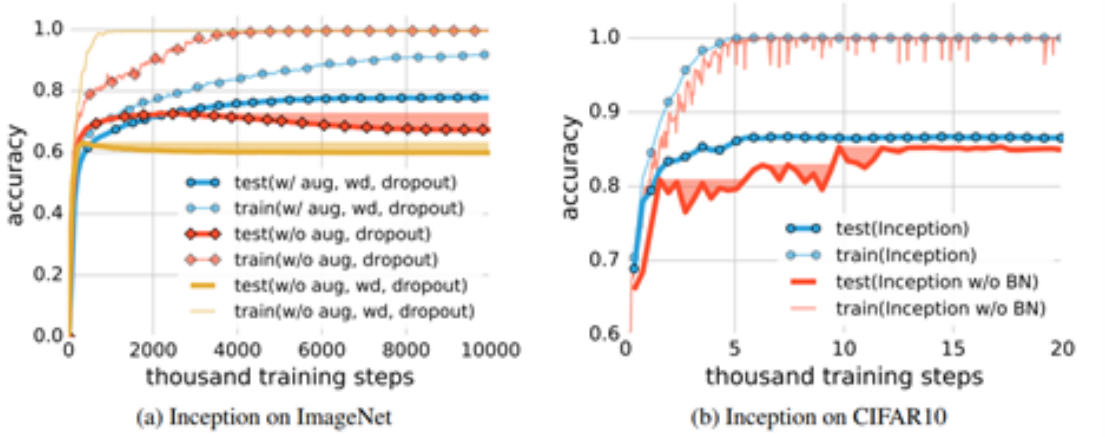

Fig. 10. Effects of implicit regularisers on generalisation performance. aug is data augmentation, wd is weight decay, BN is batch normalisation. The shaded areas are the cumulative best test accuracy as an indicator of potential performance gain of early stopping. (a) early stopping could potentially improve generalisation when other regularisers are absent. (b) early stopping is not necessarily helpful on CIFAR10, but batch normalisation stabilises the training process and improves generalisation.

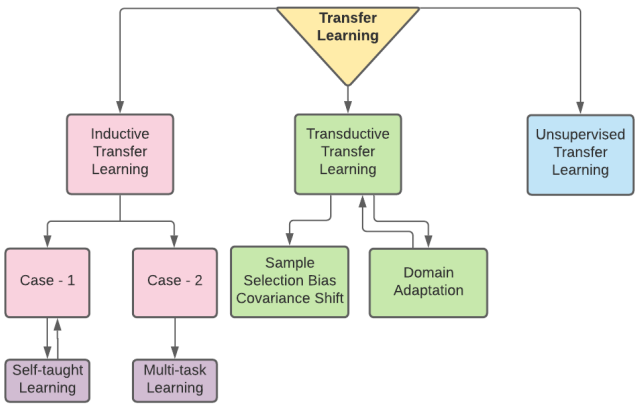

Fig. 11. Types of Transfer Learning.

"The world's most valuable resource is no longer oil, but data." [95]. However, good data is insufficient or unnecessary in various research topics, such as image segmentation, satellite, agriculture, and medical imagery. It presents a challenge of data- and label-efficient visual learning in realistic and imperfect visual environments. Thus, a critical objective of deep learning-based learning is to be as effective as possible in using even sparse, imperfect, unlabeled, and noise-filled training data to build accurate models for the real world.

Despite their empirical achievements in computer vision, deep networks frequently require large and carefully curated datasets. Data and labels are notoriously hard and expensive to acquire. For example, astronomical and scientific experiments frequently necessitate costly and high-risk data. As cost-effective data collection and preparation techniques are required for large-use cases, crowdsourcing for data labelling is often preferred. However, big data also contain private or sensitive data that must be presented to conceal its true nature.

Datasets are an important feature of deep learning, and these datasets have led to the development of important discoveries in the field. For instance, a dataset in computer vision is used by developers to acquire digital images to research, practise, and analyse their algorithms. Labeled datasets for and carefully designed training supervised and semi-supervised deep learning algorithms are typically difficult and costly to

produce because of the time required. Although the datasets needed for unsupervised learning can be either high or low-quality, they are rarely marked and often very difficult and expensive to acquire. However, the development of effective datasets has greatly contributed to the development of machine and deep learning.

In 1986, the University of California, Berkeley Office of Information Systems and Technology started a project to provide fine-quality digital photographs from their Art Museum, Architecture, and Geography Slide Library. The developers agree that this image database (eventually called Query) was the first to employ multi-user networking [96]. Until the year 1996, massive picture databases were the stuff of science fiction. At the time, storage space was incredibly limited, network performance was intolerable, and visual devices were of little use. Most of the image database creation tools had limited use in the industry.

Figure corresponding thumbnail images to a simple list of facts about the images (which is, in turn, was bound to the image list)" This was a successful search technique for both locating the right image and rapidly recognising the image on the screen.

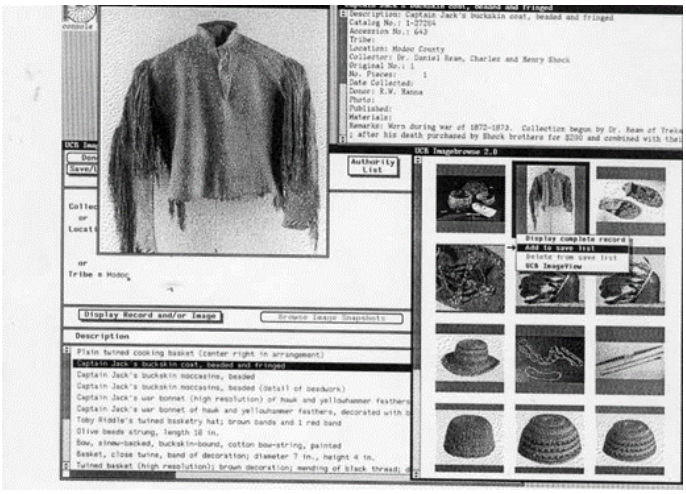

Fig. 12. ImageQuery Screendump (images courtesy of Phoebe Hearst Museum of Anthropology, UC Berkeley) Source [97].

COIL-100 [98] was collected by the Center for Research on Intelligent Systems at the Department of Computer Science, Columbia University. The database contains colour images of 100 objects. The objects were placed on a motorised turntable against a black background, and images were taken at pose internals of 5 degrees. This dataset was used in a real-time 100 object recognition system whereby a system sensor could identify the object and display its angular pose. The idea of using image databases on a wide scale didn't seem to be credible until the year 1996.

In computer vision, data annotation and dataset development are being applied at an industrial scale. Computer vision researchers use crowdsourcing platform such as Amazon's Mechanical Turk [99] to procure annotated images. The annotators are actively engaged in interpreting, filtering, cleaning, and performing this task. The term "the web-scale of things" means it should be done at an imposable magnification. The increasing size of contemporary datasets, their increasing pace, the demands made on annotators, and the widespread use of images s to depict categories are among the primary concerns for learning to sight. Efforts are undertaken to alleviate these research tensions recorded on the ImageNet website. One of the world's largest human-annotated image databases is ImageNet [100]. ImageNet presents a myriad of challenges for image classification, image description, data aggregation, image assembly, and distribution.

Yet another common technique is synthetic data use [101] [102]. Synthism refers to data that has been generated artificially rather than naturally [103]. It is commonly built by hand and used for various activities, particularly in product testing, model testing, and the development of new tools. It is useful because it can be produced when existing (actual) data cannot meet unique needs or demands. The usefulness of synthetic data in deep learning is growing with each passing day [103]. It plays an important role in developing deep learning techniques in areas where data is scarce and rare.

An application area for synthetic datasets is autonomous vehicles (AVs) [104] would revolutionise our way of life. Self-driving car simulations pioneered the use of synthetic data. Running efficient AVs at scale, however, remains to be solved. Deep learning has been a game-changer in the application areas concerned with self-driving vehicles [105] [106]. These methods can be categorised as two major frameworks: end-to-to-end driving and conventional engineering stacks [107], [108]. An end-to-to-end interpretable neural motion planner (NMP) [107] is a representative interpretable approach that takes LiDAR point clouds and HD maps as input and returns end-to-to-end motions that can be defined in the form of 3D detections. Each SDV position can be thought of as a measure of "goodness." Then the planner selects a sample set of diverse and feasible trajectories and chooses the cheapest among them for execution.

A deep learning algorithm will do whatever the training data tells it to do. If the data is bad or biased, the learned algorithm will be too. A recent demonstration is Chatbot Tay released on Twitter by Microsoft to learn from interactions with users. It started mimicking offensive language and was shut down within a short time [109].

However, not all datasets present good representations of data. Data bias is a long-known problem in machine learning as it stays in deep learning. Inappropriate outcomes are obtained if skew in data is present or if a deep learning algorithm is trained on unrepresentative data. An experiment on " Name That Dataset!" is discussed [110], where images from twelve popular recognition datasets were used. The goal was to guess which images came from which dataset. It was observed that training on one dataset and testing on another caused a big drop in performance.

Algorithmic bias [111] identifies systemic and persistent errors in a computer system that results in biased outcomes, such as this one. There are a variety of biases that may be introduced in an algorithm, including but not limited to design issues, decisions related to the data, coding errors, or unanticipated data usage that might be responsible for

Fig. 13. Unbiased look at dataset bias. Source [110].

causing a bias. Algorithmic bias occurs on search engine results and social networking sites such as Facebook and LinkedIn, and Twitter and may contribute to subtle or not-so-so-subtle unintended impacts, intentional prejudices such as racism, sexism, and ethnic biases.

Other common examples of data bias in the field are as follows: sample bias arises when a dataset does not correctly represent the model's environment. An example of this is facial recognition software being used mostly on white males [112]. The accuracy of these models varies considerably with women and non-white individuals. Selection bias [113] can be defined as the tendency to include only positive instances in a survey and leave out negative ones narrowing down the scope of data to reduce. Exclusion bias [114] arises when training data varies from real-world data. Recall bias [115] results when you inconsistently categorise similar types of data. It causes to be less accurate. Biased data may support specific populations, such as ethnic or racial groups.

The problems associated with big data, scarce data and biased data will continue shading the deep learning revolution. Overall, the last decade's progress has made deep learning algorithms capable of learning from vast quantities of data. Some issues, such as object detection and machine translation, information retrieval, text-to-to-speech, and recommender systems, are already highly scalable when trained with large data quantities. There are characterisations of personalised healthcare, robotic reinforcement learning, emotion analysis, population identification as small data problems, or big data problems as small datasets containing small information or related information groups. The ability to learn quickly in a sample-efficient manner is essential in these data-constrained situations. Collectively, these issues underscore the need for data-efficient deep learning [116] with the ability to learn in diverse domains with limited data.

Few questions are crucially related to the use of data in deep learning:

How can we avoid the use of large datasets and labelling to solve challenging real-life problems with deep learning?

How can we make deep learning algorithms free of prejudices such as racism, sexism, and ethnic biases?

How can we achieve data-efficient deep learning without harming its performance?

The sophistication of deep learning systems makes them difficult to understand. These models require previously collected data and use variables to make their decisions. However, transparency is tricky —and when things go wrong, it is difficult to correct the problem. Two aspects are important in this regard, called interpretability and explainability. Although their definitions are closely related, some works focus on interpretability while others on explainability [117].

6) Interpretability and Explainability: Interpretability is a machine learning model's ability to identify the causes and effects is an important consideration [118]. In other words, Interpretability describes the ability to understand the mechanics without being required to comprehend the reasoning behind them. Or, another way, you can predict how your programme will behave when faced with an input or the degree to which it can be altered based on different algorithmic factors. And having a knack for algorithms is just another way of seeing what's going on. Interpretability sometimes needs to be high to justify why one model is better than another.

Meanwhile, explainability is the degree to which the internal machining or deep learning method can be human-understandable [119]. The function of interpretability is a bit similar to knowing mechanics, though it does not have to necessitate it. Explainability is important in Deep Nets because the parameters, if known, allow the findings to be justified. DL models are often called black-box models because they allow a pre-set number of empty parameters, or nodes, to be assigned values by the machine learning algorithm [117].

For instance, healthcare organisations are especially concerned with making their AI algorithms accessible outside the organisation and seeking enhanced interpretability and explanation, so we must ensure we can meet these needs before any significant DL impact on healthcare is realised [120]. Apart from the more technical legal and professional considerations, improving understanding and expressiveness are also in less esoteric scenarios. Better informing data scientists and analysts about algorithms will lead to better coordination of their work with their organisation's main questions.

interpretability and Explainability: A Machine Learning Zoo Mini-tour AND Explainable AI A Review of Machine Learning Interpretability Methods Pantelis Linardatos*, Vasilis Papastefanopoulos and Sotiris Kotsiant

Interpretability is also coupled with trust [121]. To be trusted, an ML system must be able to explain its results. Two factors often determine trust: "how many times is a model correct?" and "for which examples, is it correct?". It is often useful to find causal relationships rather than mere associations. Domain-related theorising asserts that an ML system should be robust to noisy inputs and shifting domains. It is needed in social, economic, or medical decisions, for example. Readings and interpretations can be useful in identifying biases in demographic and other datasets in fact, a number of practical things that can be done y to improve an algorithm's interpretation and explainability.

The first thing is to strive for is an improved generalisation [122]. This sounds clear to the ear, but it is not an easy

task. In a lot of machine learning applications, the model sometimes feels like a tool rather than an end in itself. By improving the algorithm interpretability, you can also increase the quality of the data that is used to sustain it [123]. Similarly, plenty of implementation problems arise out of looking closely at the algorithm's different features in order to achieve real interpretability and explanation. Knowledge Graphs have been designed to capture knowledge from heterogeneous domains, making them a great candidate to achieve explanation in deep learning systems [124].

One approach to interpretability in machine learning is to be model-agnostic [125], which considers the original model as a black box to extract explanations. It can be done by learning an interpretable model on the predictions of the black-box model. LIME (Local Interpretable Model-Agnostic Explanations) is a genuine research-based method [126]. The researchers conclude that LIME "can describe [any] classifier's predictions understandably and truthfully, by learning a learnable model around the prediction." The Lime experiment simulates the model to find out by checking it. It is an attempt to replicate the output from the same phase of experimentation. However, the "black box" sense of neural networks becomes a hurdle to adoption in applications where interpretability is crucial. DeepLIFT (Deep Learning Important Features) [127] decomposes the output prediction of a neural network on a specific input. It involves a process of backpropagation to assess the contributions of all neurons in the network to every feature of the input. Shapley Additive explanations (SHAP) [128] is a game theory approach that strives to give interpretability a boost by computing significance values for each function.

Any of the problems of neural networks is finding out exactly what is going on in each layer. In the process of training, each successive layer uncovers gradually extracts higher and more abstract features until the final layer becomes the same as the input image. The first layer, in the case of image inputs, searches for corners and edges. Intermediate layers discover overall structures or elements, like a door or leaf. Those are the final few layers; they work on intricate subjects, such as whole structures or trees.

Feature visualisation provides a great aid for interpretability. One way to conceptualise the process is to ask the network to modify the input image in such a way as to evoke a certain response. For instance, if you are interested in finding out what type of image could lead to "Banana."The algorithm begins with a high randomness level but later changes to form a real photo of a banana [130], [131], [132].

Deep Visualization Toolbox [133] is open-source software that helps you understand the functioning of DNNs by feeding in an image (or a live webcam stream and studying the results together.) You can also pick neurones to display made visualisations of the visual concept that that neurone needs to focus on.

DeepDream [132] is an interoperable system that visualises the patterns learned by deep neural networks. It finds complex patterns. Similar to when a child attempts to perceive random patterns in the sky, it yields an image by first transmitting it and then measuring the gradient of the image with respect to the activation. This is done to enhance the activation patterns seen by the network and transform them into a more vibrant, dreaming picture. in order to signify the transfer of air during a time of saturation when liquid substances (e.g., sugar) become trapped and rendered solid (e.g., crystallised) (a reference to InceptionNet, and the movie Inception). Applying the algorithm iteratively and then exploring the set of items the network after each cycle gives us an infinite query. It can also be done from a random-noise image so that the final result is solely the neural network result. Source [132].

Deep neural control and perception will likely be important in self-driving vehicles. The models should have an easy-to-to-understand explanation for their unique behaviour; passengers, insurance agencies, and others should be able to understand the reasoning behind their actions [134]. The trouble with both explanation and interpretations is that they demand an investment of time during creation. It resembles to trap complexity inside greater complexity. According to one hypothesis [135], explanations that are good and are satisfying to users enable users to develop a good mental model. A good mental model will then improve trust in deep learning and AI systems in general. It is important to develop methods that can elicit mental models quickly and result in data that can be easily scored, categorised, or analysed. Few qustions are important:

How can we interpret each operation in a deep neural network precisely? How can we develop deep learning systems that can expose explanation in a human-comprehensible way? How do we evaluate the goodness of explanations of deep learning systems?

7) Security (Robustness, Safety and Reliability): The vast potential of deep learning for cybersecurity is widely recognised [136]. Internet traffic analytics can boost threat detection, reducing false alarms and network attack detection efficiency by identifying good and malicious network activity. It is possible through Recurrent Neural Networks (RNN), so these could be useful for building smarter IDPS systems. Artificial Neural Networks are useful for reacting to common online security threats such as SQL injection and Denial of Service (DOS) attacks. Neural networks could help automate discovering and exploiting vulnerabilities, as the DARPA Cyber Challenge competition [137]. Deep learning is playing a crucial role in developing cyber-physical systems (CPS) that are computer systems in which a mechanism is controlled or monitored by computer-based algorithms [138]. Examples of CPS include autonomous automobile systems, medical monitoring, industrial control systems, robotics systems in areas as diverse as aerospace, automotive, chemical processes, civil infrastructure, energy, healthcare, manufacturing, transportation, entertainment, and consumer appliances.

However, as deep networks are common in modern contexts, their design's possible security vulnerabilities are ignored. Usually, even a minor change in the input can cause the network to err with high confidence. It has created a new and complex cybersecurity threat that focuses on neural network vulnerabilities; Ian Goodfellow wrote one of the first papers detailing neural network vulnerabilities. It was dubbed adversarial machine learning and is frequently confused for generative adversarial networks [139], [140]. Poisoning attacks

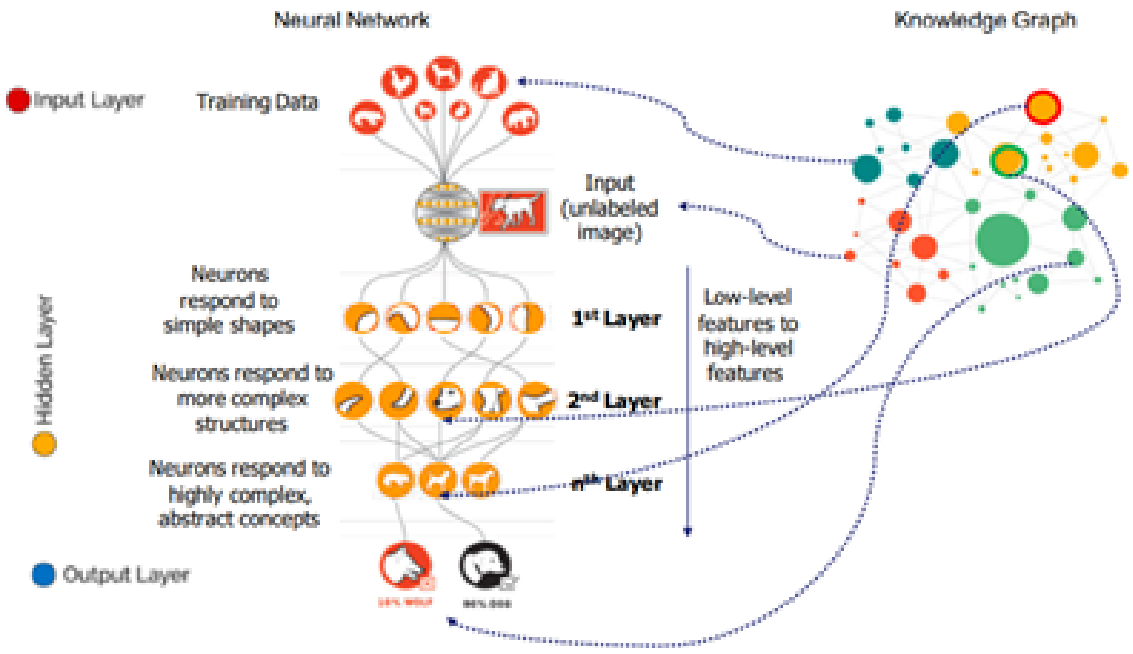

Fig. 14. The Role of Knowledge Graphs for Explainable Artificial (Deep) Neural Networks. Source [124]

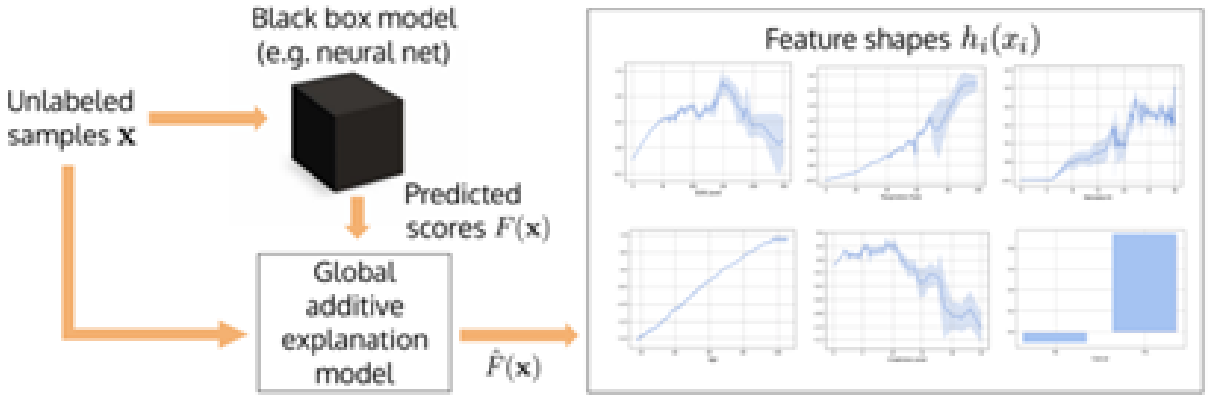

Fig. 15. Given a black box model and unlabeled samples (new unlabeled data or training data with labels discarded), model distillation is used to learn feature shapes that describe the relationship between input features and model predictions—source [129].

vulnerabilities harm the algorithm's learning capability slightly more devious techniques like the tried and true methods of attack evasion do not work on participants who have implemented online learning like that.

White box and black box attacks are two general kinds of attacks [140]. In white-box attacks, attackers have access to the whole network. Thus, they can understand the structure of the network. Knowing the network's layout will allow them to identify the most effective attacks and reveal vulnerabilities pertinent to the network. In black-box attacks, the intruder does not have direct knowledge of the network. If we can test as many samples as we want on the network, we can design a network to get a bunch of training data as output. Using these labelled training samples, we can then build a new model that has the same performance. For neural networks, the unknown structure can be regarded as a black box.

Adversary attacks often used in learning systems that use neural networks and deep learning. Adversarial approaches can be used to modify a music file to instruct a specific audio system to operate in a particular way when played. The human player wouldn't know that the file includes secret commands. The researchers at UC Berkley created a proof-of-concept that, in which they only adjusted the audio files, could fool an AI to transcribing a machine into thinking something that sounds

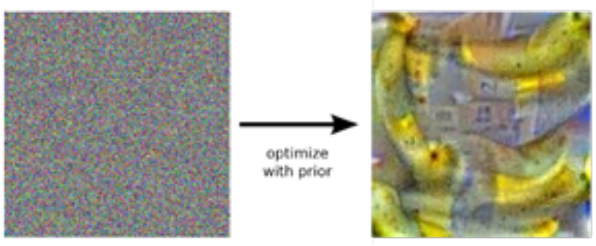

Fig. 16. Seeing Banana response Source [132]

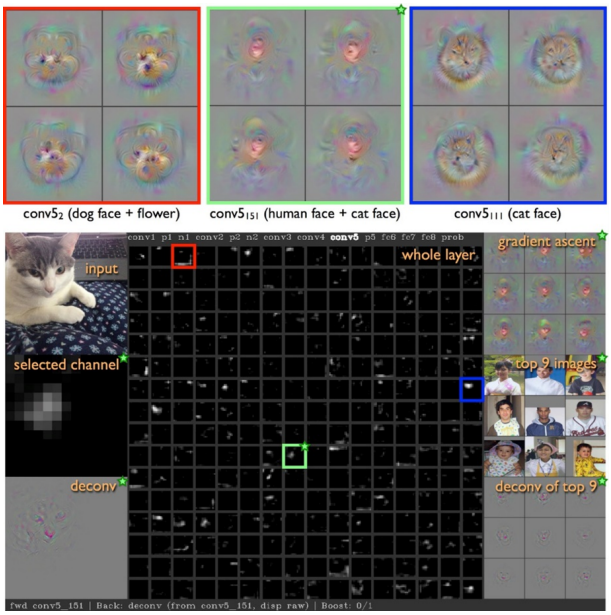

Fig. 17. Facial depiction of Ms Freckles using the Deep Visualization Toolbox. Source [133].

different to human ears [142], [143].

In an effort to combat these attacks, numerous countermeasures have been attempted [144]. Data can be tweaked in an adversarial manner to make a model stable in the face of random perturbations (which is a special case of variability). Using a shield against adversarial threats is done using adversarial examples. It is based on actively generating ”adversarial” instances, changing their names, and introducing them to the training process. The new network is then trained with this revised training collection, which will help the network resist attacks. For example, feature squeezing [145] is suggested to detect adversarial examples. Density estimates and Bayesian uncertainty estimates are proposed [146] to detect adversarial examples. Adversarial examples are well suited for improving the deep learning system protection, as they deal with specific problems that can be dealt with easily and are complex enough to warrant a dedicated research effort. Advances in the development of adversarial samples have been made, but still leave many issues and barriers to overcome.

In general, regularisation is always providing some solution. It does a decent job of smoothing the boundary class assignments and helps reduce the effects of strategic noise injection [138]. Authors in [147] apply regularisation using the Frobenius norm of the Jacobian of the network leading

Fig. 18. Generated from purely random “noise”, using a network that was designed by the MIT and AI Laboratory Look at our Inceptionism gallery for higher resolution pictures. Source [132]

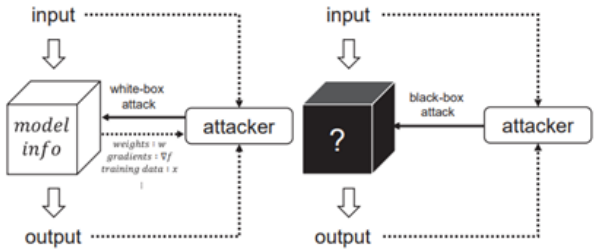

Fig. 19. illustration of white and black-box attacks. Source [141]

to enhanced robustness results. CleverHans [148] is a library that is designed to assess how easily adversarial examples can exploit machine learning systems.

However, many defensive techniques are only successful against certain forms of attacks but struggle to cope with unknown attacks. Methods for defending against adversarial examples are few and far between. A different approach is to finding threats is using computer security and algorithms for computer security rather than machine learning to detect artefacts.

The privacy concerns for deep learning were currently established, and numerous threats were suggested [141]. Intellectual propriety and sensitive training data sets of a DL model (e.g. parameters, architecture) is referred to as DL privacy. The attacks that violate the model’s privacy belong in two categories: model extraction and model reversal attack [150], [151]. The opponent seeks to double the parameters or hyperparameters of the model deployed to offer cloud-based ML services, which compromise Dl algorithms’ confidentiality and the intellectual property of services providers in model extraction attacks. The adversary tries to gain classified information using usable information in model inversion attacks.

Four mainstream technologies for privacy protection from DL are currently available, namely differential privacy, homomorphic encryption, reliable multi-part computing and a trustworthy running environment [149]. Differential privacy is intended to prevent an opponent from finding out if a specific case had trained the target model. The homomorphic encryption and stable multi-party computing system concentrate on protecting data privacy. The trusted environment aims to build a protected and isolated environment with hardware to secure training code and sensitive data.

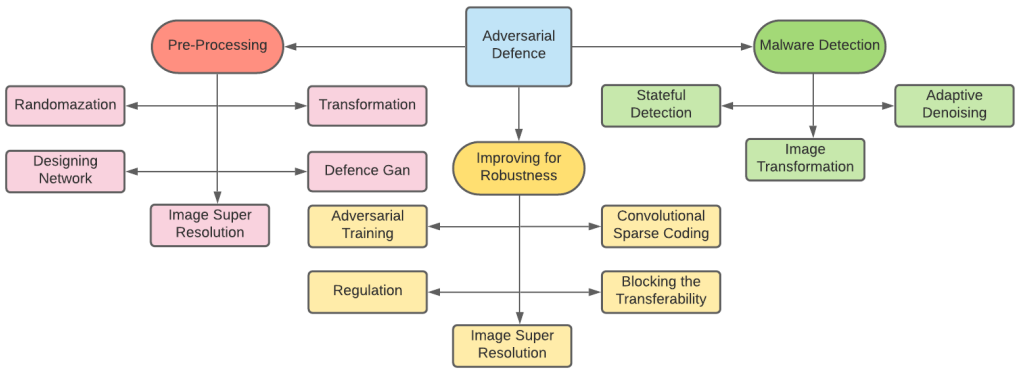

Fig. 20. Taxonomy of Adversarial Defences.

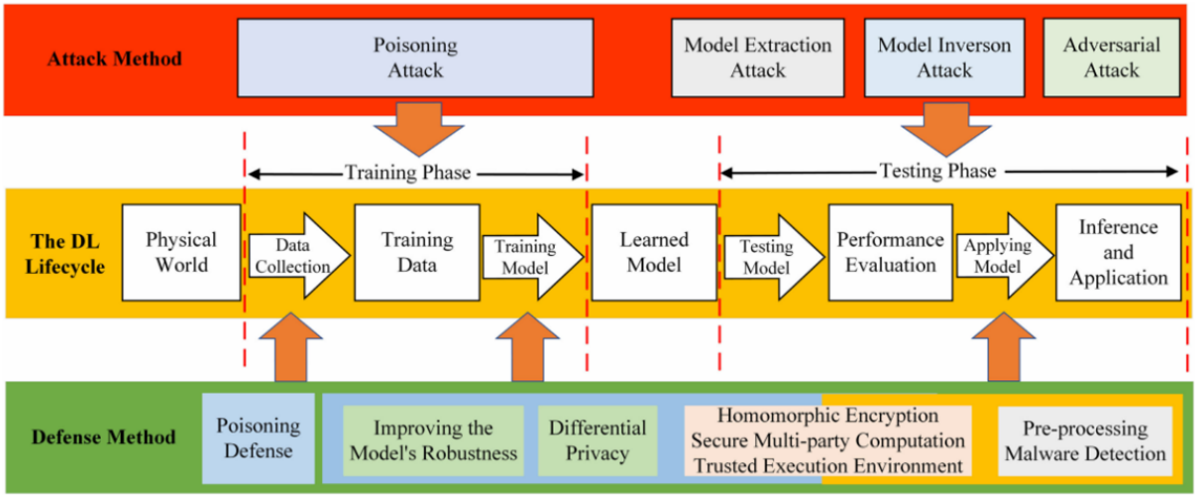

Fig. 21. Overview of attacks and defences in DL. Source [149]

The approach to the integration of privacy into device architecture by design is known as privacy by design, which Ann Cavoukian implemented in 2009 [152]. This definition is included in the European GDPR [153]. The confidentiality learning system uses the concept of privacy. Google recently developed a specification for machine learning to protect training data privacy [154]. The distributed architecture of distributed federated learning systems, where model training takes place locally, is then sent to a central server for aggregation.

Federated learning is a powerful framework that machine learners can use by default to collaborate with decentralised data [155]. The federated architecture initial focused on device-based, mobile user-custom-driven learning. One example is the Gboard application, which predicted that typing with the Federated Recurrent Network (FRNN) model is done easily based on the typed text [156]. In the literature following the initial design, several changes are proposed. This includes horizontal and vertical, federated learning, federal transmission, federated domain adaptation, adversarial learning federated, enhanced communication and security protocols and made federated learning more customisable. It appears in machine learning as a promising topic for research. Refer to the subject reviews ([157], [158]) for interested readers.

Privacy by design The theoretical framework proposed is the extension of the privacy framework for machine learning design to include the concept of incorporating different privacy into its system design (Cavoukian et al., 2009). Figure presents a visual overview of the architecture (dPbD) [159] structure based on differential privacy for federated learning. The proposed structure includes four quadrants and four key connections, defining the most critical factors in defining differential privacy in federated learning. These include the degree of privacy, random noise, global sensitivity, and a diverse federated private learning system. It includes the number of client nodes.

As differential privacy guarantees anonymity and indistinguishability in terms of a privacy budget (epsilon)—the smaller the budget, the stronger the confidence in privacy. It is a level of privacy. It is significant characteristics of differential privacy by design as it guarantees and provides a quantitative notion of privacy compared to the unclear concept of privacy in privacy by design framework. The level of privacy defines system robustness against attacks. Similarly, adding more

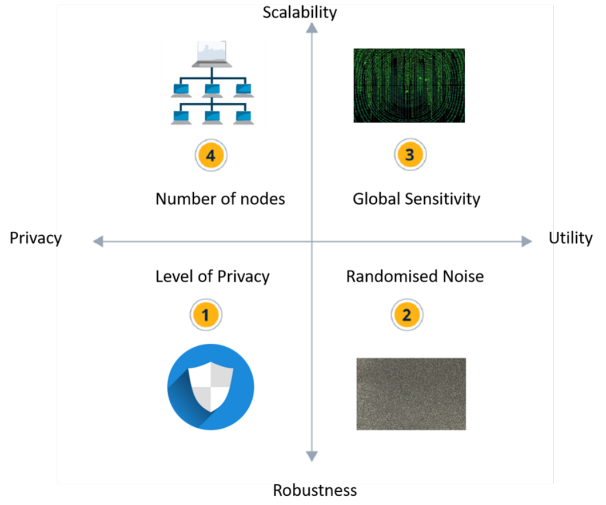

Fig. 22. A visual overview of the architecture: Differentialprivacy by Federated Design (dPbD). Source [159] [198]

noise as part of differential privacy can reduce system utility and robustness. The amount of randomised noise is thus an important factor to balance the trade-off between privacy and utility. On the other hand, scalability requires an increase in the number of nodes and robust aggregation effectiveness. Robust aggregation depends on global sensitivity. A small value of global sensitivity requires that all the clients use sufficient local datasets for training and the type of aggregation function. Similarly, low global sensitivity functions are preferred for better performance and utility like sample mean and covariance matrix.

DeepLearning has also contributed towards increased surveillance and security in our modern society. The use of cameras and drones are in abundance. On one end, it provides security, and on the other end, it has harmed our privacy. At the same time, adversarial learning has encouraged a battle between adversaries and defenders. It raises many concerns for the future. Answering these assaults and defensives gives rise to several questions:

How can we develop a generalised defence against all possible adversarial attacks? How can we solve the security vs privacy trade-off effectively?

TABLES: known attaches,vunrabilities, solution proposed

8) Inclusive and responsible DL: The development of fairness in data and deep learning algorithms from the ground up through conception is crucial to secure, reliable AI systems [160]. Though precision is a metric for assessing a machine study model accuracy, fairness provides us with a way of understanding the practical consequences of using the model in real-world situations. Fairness is the comprehension process that data entails and ensures that the model delivers a fair forecast for all demographic categories. It is important to use fairness analysis during the entire ML process rather than thinking of fairness as a separate programme, ensuring the models are consistently reassessed from a fairness and inclusion perspective. It is particularly important as AI is used in critical procedures, like credit application examinations and medical diagnosis, which impact a broad range of end-users. When the first iteration of the model has been deployed, it is best to collect input on the results of the model and take steps to make the next round fairer.

The unfairness of DNN can usually be classified into two groups from a computational perspective: discrimination in the result and inequality in the consistency of prediction [161]. Discrimination involves the phenomenon of unfavourable treatment of persons by DNN models because of the participation of certain ethnic groups. The prejudice resulting from the prediction may be traced back to the contribution. Although a DNN model does not take protected characteristics specifically for input, e.g. race, sex and age, it may cause discrimination in prediction.

The success of deep learning raises issues of justice, openness, privacy and more: subjects that are often grouped in the sense of ”algorithmic accountability”. TensorFlow World released a beta version of Fairness Indicators [162], a suite of tools that allow routine calculations and visualization of fairness measurement for binary and multi-class classifying. Fairness indicators can allow developers to make better decisions about how to deploy models responsibly by creating transparency reports such as those used on model cards [163]. Detecting and mitigating unfair prejudice and historical discrimination that deep learning models learn to emulate and spread is crucial for its long-term use. It is important for the computing community to take a more meaningful approach to the technology’s social effects that the field creates and strives for a more just, rigorous, and accountable field. Perhaps we could call ”Fairness by Design” a fair and accountable approach to software design and implementation [164] [209].

Trust and responsibility in deep learning [165] can be defined in a multitude of ways. Industry-standard definitions are useful for those disciplines that use standard practises, e.g., automotive and avionics. To begin with, trustworthiness typically exists as an issue in two contexts: certification and clarification. The product validation process is conducted before product implementation to ensure that it performs properly (and safely).

Verification problems usually have high computational complexity, such as being NP-hard, when the properties are simple input-output constraints [166]. This, compounded with the high-dimensionality and the high non-linearity of DNNs, makes the existing verification techniques hard to work with for industrial-scale DNNs. The goal of testing DNNs is to generate a set of test cases that can demonstrate confidence in a DNN’s performance, when passed, such that they can support an assurance case [167]. Existing approaches to verifying networks largely fall into the following categories: constraint solving, search-based approach, global optimization, and over-approximation; note, the separation between them may not be strict [168].

However, there are various possible failure modes, which may be discovered only after the model is implemented: models may fail to achieve their recorded accuracies, make arbitrary decisions, or provide unappealing predictions to improve generalization to new data domains. To construct systems that work in the face of novel, even hostile, inputs,

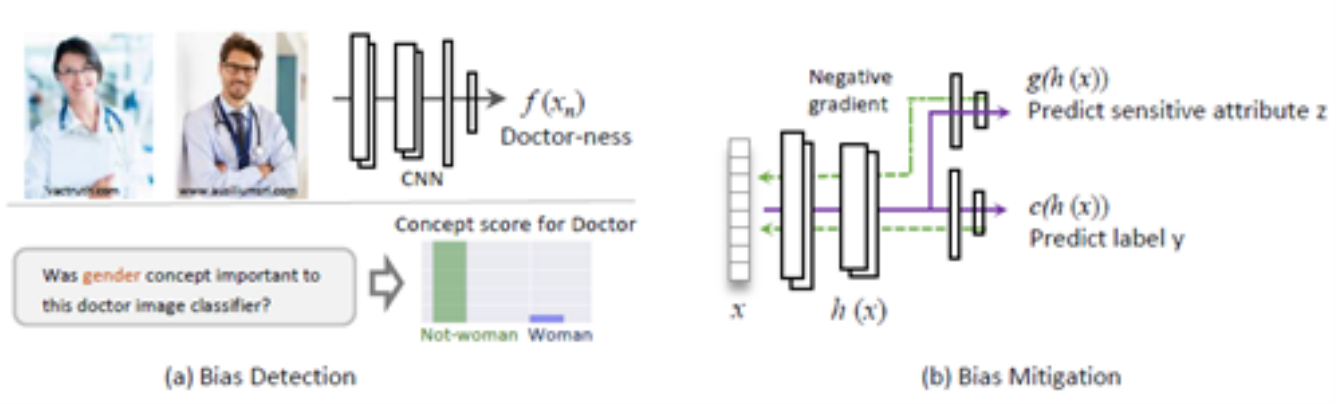

Fig. 23. Global interpretation for detection of discrimination. Results show that this CNN has captured gender concept, and the not-woman concept would significantly increase doctor prediction confidence of the CNN classifier. Thus it indicates the CNN's discrimination towards woman. (b) Adversarial training for mitigation of discrimination. The intuition is to enforce deep representation to maximally predict main task labels, while at the same time minimally predict sensitive attributes. Source [161]

how do we go about doing so?" autonomous systems must be and can be developed in such an automated fashion on a large scale, but also in a difficult or infeasible manner Protection of emerging applications of AI such as self-driven cars and surgical assistants dependability needs attention to these issues.

There is a special case of self-driving vehicles and their reliability. It will take us a few miles to obtain the desired rate of reliability. Strictly specified by the ISO 26262 reliability scheme [169], the safety rating "Automotive Integrity Level D" (ASD) is applied to self-driving vehicles accordingly. In order to guarantee human safety and promote public acceptance of the nascent self-self-driving technology, stringent guidelines are required. Even though there have been some big improvements, many people still feel it is dangerous.

In 2016, a self-driving car suffered a visibility failure [170] because of the lack of clear contrast in the image between a brightly lit sky and a black backdrop. A better camera that is said to be able to take on a more rapid light sensitivity may Even though a more realistic and much less prone to failure is failure is prevention. Complementary sensor technologies, such as radar, LIDAR, and camera, are helping the industry move towards improved redundancy. Two or more sensors should be deployed at all times when autonomous driving is engaged to ensure the effectiveness

Accountability and justice are related principles. Organizations such as the US Food and Drug Administration (FDA) are now proposing regulatory mechanisms to standardize AI, and ML use specifically for medical devices and to verify them [171]. It is a major step that contributes to the enforcement of a quality bar for the work of data scientists and requires increased responsibility for the models generated by those experiments.

Ethics is, of course, one of the most critical issues in the last couple of years in machine learning [172]. One application area is healthcare. The existing policies and ethical standards related to deep learning lag behind AI's progress in health care. Although attempts have been made to participate in these ethical talks, the health community is still not aware of the ethical complexities of emerging AI technology. A rich debate is therefore anticipated that will benefit considerably from physician feedback, as doctors will possibly interact with AI in their daily work in the near future.

Another ethical concern is related to the spread of deep learning-based fake images, photographs, videos and content. Everything we hear, see, and read is difficult to believe now. Deep adversarial networks (or GANs) [173] has paved the way for deep adversarial networks. Deepfakes (a portmanteau of "deep learning" and "fake") [174] are fraudulent media where someone is secretly substituted in an actual picture or video. Although content faking has been performed before, these techniques exploit powerful deep learning and artificial intelligence algorithms to generate visual and audio content or videos with a high potential to deceive (GANs). In addition to audio faking, celebrities and political figures may be replaced with "voice prosthetics" in order to portray them in "mockumental" or satiric ways [175]. Detecting deepfakes is difficult and a major ethical concern for modern society [176].

Fig. 24. Gradual improvement in GANS and thus, Deepfakes Creation. Source [177]

Deepfakes are widely used for malicious purposes, making it easy for virtually everyone to produce deepfakes these days. Until now, various approaches have been proposed to detect deepfakes: as such, there is a struggle between malicious and beneficial applications of deep learning [177]. However, deep learning community is actively exploring ways to detect deepfakes [178]. How do we learn a value function for

systems with complex desiderata, which catches and balances all relevant considerations? How does a machine behave if its value function is uncertain? Should we ensure that the beliefs of people who use a system are reflected? In future, value-sensitive architecture is a big trend. It requires a value-sensitive-design (VSD) approach for deep learning systems [179].

As machines get smarter and more and more of our machines enquire, how do they handle themselves and how do they see themselves in society? Once computers can indeed mimic emotions and behave like human beings, how are they to be governed? Should machines be considered as people, animals or inanimate objects? To this extent, to what extent are we responsible for the machines themselves over the people they are meant to control?

The following questions are important in this perspective:

Can we design something resilient to novel or potentially adversarial behaviour? What methods exist for dealing with model mistakes or mistakes in training data? If we have device behaviour and environmental data, what else could we track?

Tables: Privacy techniques,federated techniques, deeo fakes ,ethical approaches

9) Unlocking Creativity: Creativity is seen as essential to human cognition and has recently gained widespread attention. Creativity is referred to as the process of producing novel and useful concepts and thinking to include cognitive abilities of novelty and value (the new idea or object must be praiseworthy) [180]. It is not a "gifted" personality, nor is it present only in an isolated genius. Creativity is considered a very complex ability that demands many qualifications. Human intelligence is rooted in daily skills such as the association of thoughts, reminding, analogical thought, problem-solving, self-critical thinking, and searching. It is not only a cognitive process (invention of new ideas), but it is also dependent on one's motivation and cultural background, and personality traits.

Until recently, it was assumed that only humans could produce and appreciate art. With all the advances in computational innovation and increased use of artificial intelligence in creative fields, this assumption is increasingly being called into question.

Overall, creativity is typically defined as having two elements: originality and meaning [181]. Valuable ideas that are novel and interesting are considered creative. "Novel" means two different things in this context. The idea can be a novel idea only to the mind of the individual who conceived it. It can also not have happened before in history. The first one can be known as P-creativity (P for psychological), the latter as H-creativity (H for historical) [182].

A "fine-grained categorisation" was suggested by Kaufman and Beghetto [181] as it was categorised as mini-c, little-c, pro-c, and Big-C creativity. Mini-c and little-c Creativity are a form of everyday creativity like creative processes that generate tangible outputs, or a novel way of interpreting such sensations, such as new experiences. Big-C creativity generates creative outputs that have a considerable impact on a field and are often connected with the notion of genius. Pro-c creativity generates outputs that are recognised as being novel to a domain without considerably revolutionising the domain.

Boden [182] addresses three varieties of creativity: combinatorial, revolutionary, and transformative. These three mental strategies work in a conceptual space that addresses a person's perspective on the issue. The first category includes novel (unlikely) variations of common concepts and informal (complex) concepts. This. "combined" type of creativity is referred to as combinatorial creativity. The examples are a large amount of poetic imagery based on a shared underlying structure. Linguistic similarities are often discussed and researched thoroughly for rhetoric or problem-solving. The second kind, "exploratory," entails generating novel ideas through the exploration of formal, conceptual spaces. It sometimes leads to systems ("ideas") that are novel and unpredictable. However, it is immediately clear that they fulfil the canons of the thinking style in question. The third type of creativity is "transformational." That involves transforming certain (one or more) dimensions of space, allowing for the formation of previously unimaginable structures. The more basic the dimension at stake and the more strong the change, the more unexpected the new ideas would be.

Researchers have tried to evaluate creativity by quantitative and qualitative means. For instance, in one of the recent work [183], the creativity of a system can be assessed as follows:

EQUATION []

Creativity is proportional to naivety (N), novelty and distance of connections (D), evaluative ability (V), and efficiency (E).

The historical notion of creativity is often linked to visual creativity [184]. Visual creativity is an essential aspect of creativity because it can produce artifacts with novel and useful visual forms, which is important in many fields such as photography, painting, and sculpture. Humans have created visual art across documented history and throughout cultures. Visual artwork, such as drawings and sculptures, has been associated with human intellect since the earliest records of human activity. Human ancestors created objects depicting actual and imaginary beings, such as animals or deities, tens of thousands of years ago. They formed complex images with more abstract meanings, such as sets of lines and shapes. The pervasiveness of art through cultures and history is an enthralling feature of human cognition. One important aspect of art, for example, is that it must be formed in the artist's mind to transform the artist's intentions into a tangible form [185].

The neural bases of creativity, particularly artistic creativity, have become a hot topic in recent years [186] [187]. Several neuroimaging experiments have been conducted to investigate the neural basis of visual imagination. However, little is known about the relationship between cortical structure and visual imagination at this time. The findings [187] also indicate that multiple brain modules are required to come up with creative ideas rather than a single "creativity module". Creativity is considered as the joint function of two competitive types of cognitive processes: executive control and spontaneous thinking [188]. The processes of Executive control regulate mental resources. In contrast, spontaneous thinking reduces the continuous stream of internal processing and association between the flow of thoughts, emotions, images and sounds.

Spontaneous thinking generates new ideas, while executive control allows for the selection and evaluation of ideas [188].

One goal of deep learning and artificial intelligence is to make computers and machines capable of doing stuff that brains do, which piques our interest in real brains and neural science [189]. Perception has historically been one of those fields that we believe the computer will never achieve: the mechanism by which objects in the environment (sounds and images) can be converted into ideas in mind — this is important for our brains [190]. The inverse of experience is creativity: transforming an idea into something new and valuable. In recent years, work on machine cognition has unexpectedly intersected with the worlds of machine creativity and machine art. Michelangelo's perspective on the dual relationship between perception and creativity is that we construct by perceiving, and perception is an act of creativity [191].

Logic inference is a form of abductive reasoning that begins with an observation and then attempts to find the most straightforward and likely explanation. It turns out that we can let the system generate/project object images for us by using the same error-minimisation method and the network we trained to recognise the object. Alternatively, you can start with a non-empty canvas (initial x') and find and generate x on that canvas. The model can find a lot of the object's image x that you're looking for in the initial x' image, such as this one: Perception and imagination are intertwined in recent work. As true vision only takes place in the brain, visual imagination can be considered as a hallucination or a dream. The DeepDream network [130], for example, provides a systematic way to "parametrically" monitor certain characteristics of the images generated in this way.

Fig. 25. Moonage Daydream: art created by Deep Dream. Photograph: Deep DreamSource [130]

Therefore, to comprehend the effect of deep learning on human creativity, we must first comprehend the distinctions between how AI and humans produce new art. Deep learning models that generate new data that is close to the training dataset are known as generative deep learning models. They are taught to approximate the training data's underlying latent probability distribution. This distribution is sampled to produce new outputs. Variational Autoencoders (AEs) [192], PIXEL RNN [193] and generative adversarial networks (GANs) [173] are the most well-established deep learning

Fig. 26. Google's Deep Dream generated this picture. Google Image Source [130]

architectures in this family (GANs). Autoencoders are made up of two parts: an encoder that maps the input data to a latent lower-dimensional representation and a decoder that reconstructs the original input from the latent encodings [192]. In October 2018, a portrait of Edmond Belamy sold for nearly $432,000, at a 45-times-increased price .Since 2017, an AI wrote a continuation of the Harry Potter books that studied all seven volumes of J.K Rowling's writing. Taryn Southern's new album was dubbed a self-made masterpiece [194].

The argument is that the so-called artistic content created by deep neural networks is often done in the style of a specific artist, so the content is never truly authentic. All great artists, including Dali, Louis Armstrong, and even Shakespeare, started practising their craft by imitating and building on the work of others. Perhaps imitation is a prerequisite for innovation. Gatys, Ecker, and Bethge [195] showed that a 19-layer VGG-Network [196] trained to recognise objects learns representations that can be used to differentiate content from shape, as well as to create arbitrary style and content combinations.The term "content" refers to "what" is represented and is associated with the subject matter, semantics, or indexically referenced subjects, while "style" refers to "how" subjects are made by the use of media and techniques that represent the production process or individual viewpoint. Various modifications of GANs now exists. If additional information is added on both the generator and the discriminator,, the model can be extended to a conditional model. Auxiliary data, such as class labels or data from other modalities, may be represented by y. One interesting GAN called the CycleGAN [197] is illustrated by photographing landscapes in the visual styles of Monet, Van Gogh, Cezanne, and Ukiyo-e.

DALL-E [198] is a cutting-edge artificial intelligence neural network that uses text prompts to produce images. OpenAI chose the name DALL-E as a nod to Salvador Dali and Pixar's WALL-E. It creates pastiche images that represent Dal's surrealism, which combines dream and imagination with the daily rational world, as well as NASA paintings from the 1950s and 1960s and Disney Imagineers' work for Disneyland Tomorrowland. DALL-E is a surrealistic and animated film. DALL-E is a 12-billion-parameter variant of the GPT-3 natural language processing neural network, which

Fig. 27. Images of landscapes as an example of style transfer from famous painters. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks is adapted from the paper Unpaired image-to-image Translation Using Cycle-Consistent Adversarial Networks. Source [197].

has 175 billion. Even though AI's understanding of human language is minimal, deep learning has found some fascinating use cases in assisting skilled writers in the field of literature. Natural language processing and generation (NLP/NLG), a branch of AI that lets computers analyse and generate human text, is most likely the technology used. NLG is the polar opposite of NLP. "Whereas NLP is concerned with extracting analytic insights from textual data, NLG is concerned with synthesising textual content by integrating analytic output with contextualised narratives," according to Gartner.

GPT-3 " [199] learns" by recognising patterns in data gathered from the internet, such as Reddit posts, Wikipedia articles, fan fiction, and other sources. GPT-3 can perform a variety of tasks without any additional training, including creating convincing narratives, generating computer code, translating between languages, and performing math calculations, among other feats, such as autocompleting images. OpenAI has improved GPT-3 with DALL-E, focusing on and extending the manipulation of visual concepts by language. Consider the following text prompt: "An armchair in the shape of an avocado. An armchair imitating an avocado," The following images are generated as a result of this:

Fig. 28. The prompt in the text "A female mannequin dressed in a black leather jacket and gold pleated skirt" gives the following results.

The field of Musical Metacreation (MuMe) [200] has produced impressive results for both autonomous and interactive creativity. Musical Metacreation (MUME) is the process of endowing machines with musical imagination using methods and techniques from artificial intelligence, artificial life, and machine learning, which are also influenced by cognitive and life sciences. In concrete terms, the field brings together musicians, practitioners, and researchers who are interested in developing structures that identify, learn, represent, write, complete, accompany, or interpret musical material autonomously (or nteractively).

The creative integration of language and images in an intuitive way has led to many interesting applications areas such as AI for the food industry. Recipe1M+ is a large structured corpus of over a million cooking recipes and 13 million food images. Recipe1M+ [201] enables high-capacity models to be trained on linked, multimodal data, a recipe-image embedding that produces impressive results on an image-recipe retrieval task.

AI is not trying to reproduce the human mind; What it cares about are the approaches to engaging with humans that nurture their imagination. The true benefit of deep learning and other advancements in the AI industry would be the enhancement of human capabilities. In reality, neural networks and deep learning are likely to make it easier for more people to become innovative. Deep learning can help artists find fresh ideas and speed up their creative process on a more professional level. It is called augmented creativity [202].

Originality and innovation are needed for creativity, but so are relevance, importance, and significance [203]. And it's this sense that has enabled imagination to take us beyond animals, and that continues to set us apart from algorithms. It's easier to think of new ideas than it is to make them work. What is difficult is trying to insert creativity into living and feeling humans in a pre-existing, breathing system of thought and feeling. In addition, it needs familiarity with the setting, as well as meaning that is hard for machine learning and AI to do at this moment in practice.

A question that presents a creative solution to a problem is more useful than one that only presents a creative response. When creating a painting, song, book, or any other piece of art, your background, culture, politics, and religion can all mix in with emotions and feelings. Overall, It's simple for AI to create something new on its own. But coming up with something new, unexpected, and useful is incredibly difficult. It raises various questions related to artificial creativity.

How van we develop deep learning systems that do more than augmenting creativity? How can we link meanings and value to artificial creativity? How can we develop creative Ai that solves out of the box problems?

10) Towards Artificial General Intelligence (AGI): Artificial General Intelligence (AGI) refers to artificial agents/programs' ability to display human-level proficiency in reasoning about and performing tasks in their environment [204]. It can be stated as the hypothetical capacity of an intelligent agent to comprehend or learn any intellectual activity that a human can (AGI). Strong AI, absolute AI, and general intelligent behaviour are all terms used to describe AGI. Computer programmes that can experience sentience, self-awareness, and consciousness are referred to as "big AI" by some academic sources [205]. AGI is thought to be decades removed from today's AI.

The most difficult problems for computers are referred to as "AI-complete" or "AI-hard" informally, meaning that solving them requires the general intelligence of humans, or powerful AI, which is beyond the capabilities of a purpose-specific algorithm [206]. General computer vision, natural language

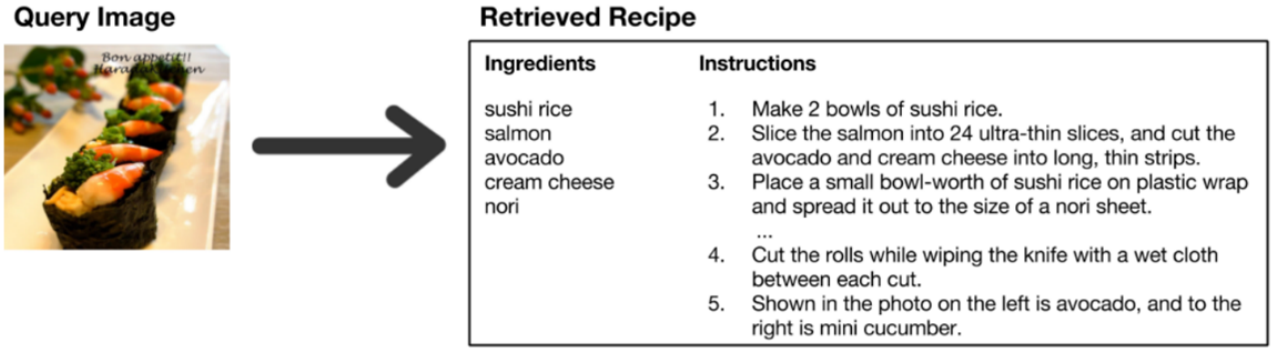

Fig. 29. Sample recipe generated by deep learning on Recipe1M+ [201] . This data set offers various challenges that can innovate food industry.

comprehension, and coping with unforeseen situations when solving real-world problems are all thought to be AI-complete problems. AI-complete problems can't be solved with today's computer technology alone; they need human computation [207].

There are two possible approaches to designing artificial general intelligence (AGI): computer science and neuroscience. These two methods depend on separate and incompatible platforms due to fundamental differences in their formulations and coding schemes, slowing the progress of AGI. It would be ideal to provide a general framework that could support both current computer science-based deep neural networks and neuroscience-inspired models and algorithms. A group of researchers created the Tianjic chip [208], which combines the two approaches to build a hybrid, synergistic platform.The Tianjic chip has a multi-core architecture, reconfigurable building blocks, and a seamless dataflow with hybrid coding schemes that can handle not only machine learning algorithms based on computer science but also circuits inspired by the brain and a variety of coding schemes for demonstrating real-time object detection, tracking, voice control, obstacle avoidance, and balance control in an unmanned bicycle system.

Reinforcement learning strategies enable an agent to comm e.unicate with the environment to maximise reward, which is a subset of general intelligence's larger problem. AlphaGo [209], [210] success story based on policy network and value network has created an account based on machine intuition that has beaten human potential. Deep Q-learning [211] and deep meta Q-learning [212] have the potential to reach a certain level of intellect for machine intelligence. However, often training of this reinforcement learning system is slow and time-consuming. It takes millions of self-play games to become able to play with human. It raises questions about the learning strategies in deep learning. However, although individually capable of performing the tasks for which they have been programmed, AI and deep learning systems lack a broad understanding of the environment. On the other hand, a professional player can modify his playing style after learning about the changes made by a new update. He can also learn a new game with minimum effort. AlphaGo Zero [209] has been designed to extend this adaptive capability to the machine.

Recent advancements in deep learning, AI, and robotics have enabled us to better sensory perception, natural language understanding and generation, fine motor control, and augmented creativity. Intelligent systems still lack human sensory perception, emotional intelligence and social engagement. Human is good at interpret feelings and reflect emotions; it's difficult to believe that machine empathy is within reach. Commonsense reasoning allows people to make presumptions regarding ordinary matters while it seems a hard task for machines at the moment [213].

Three organisations are leading AGI research; OpenAI aims to achieve or promote artificial intelligence (and ensure that it is responsibly used). Similar targets are being pursued by Google's DeepMind and the Human Brain Project [214].

OpenAI demonstrated a robotic hand that was completely trained in simulation to manipulate objects into different orientations in mid-2018 [215]. Even though the tasks seem to be straightforward, the most important thing it accomplished was the ability to do well in unfamiliar situations despite never having been explicitly trained to behave in such cases. This was accomplished using Domain Randomisation [216], a training technique that enabled the system to recognise the environment's key features and generalise to new situations.

For tasks like these, we'll need to create an agent with capabilities spanning all AI science areas, from Natural Language Processing to computer vision, problem-solving, and gameplay. Simultaneously, we need A real-time 3D engine with a spatial environment, in combination with a physics engine, to create a self-contained ecosystem that closely resembles the real world. Gaming and graphics could be a playground for AGI. The Neural MMO [217] — a simulation environment focused on Massively Multiplayer Online Role-Playing Games (MMORPGs), including World of Warcraft and Runescape — is one step OpenAI has taken in this direction.

The majority of current deep learning research is based on agents that scale to their surroundings. It necessitates the development of more sophisticated algorithms that allow agents to maximise their ability in their environment. This necessitates a clearer understanding of the fundamental tasks that an agent must perform in its environments, such as exploration and memory management and solutions for them. Environments that scale to the real world are essential because as agents improve, they will reach a point where their surroundings constrain them. To avoid this, simulations that

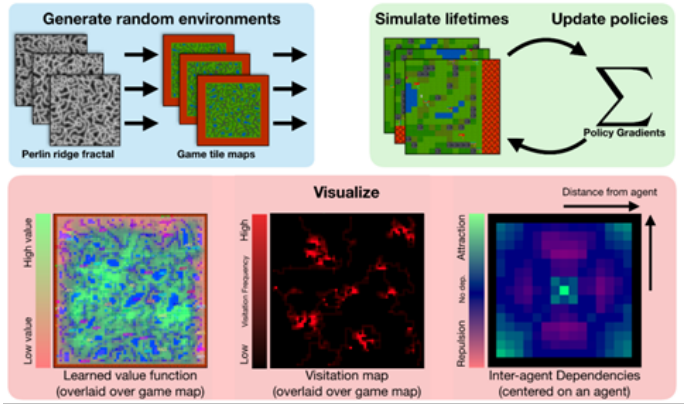

Fig. 30. The Neural MMO environment that can be constructed. Agents must explore to find fertile lands, then fight for them—source [217] [270].

are better approximations of the real world must be created.

In more specialised applications, object recognition, language comprehension, and manual dexterity can be mastered to a reasonable degree in the medium term to solve particular use cases. Few machine evaluations are interesting to consider. In Wozniak's The Coffee Test [204], a computer must reach an ordinary home and figure out how to make coffee: locate the coffee machine, locate the coffee, add water, locate a cup, and press the right buttons to brew the coffee. In Goertzel's The Robot College Student Test [204], if a computer enrols in a university, takes and passes the same classes as humans, and graduates with a degree, it can pass the test. The Employment Test [218] reroutes from a computer performing at least as well as humans in a similar job to a machine performing at least and humans in the same position.

Other facets of the human mind, apart from intellect, are relevant to the idea of powerful AI and play a significant role in science fiction and artificial intelligence ethics: It includes consciousness (the ability to have subjective experiences and thoughts), Self-awareness (the ability to recognise oneself as a distinct entity, especially to recognise one's thoughts), Sentience (the capacity to subjectively "see" thoughts or feelings) and Sapience (the ability to gain wisdom) [219] [220] [221].

Artificial Consciousness (AC), also known as computer consciousness (MC) or digital consciousness [222], is a branch of AI and cognitive robotics. Artificial consciousness theory aims to "define what would have to be synthesised if consciousness were to be found in a machine" [223]. Though there are challenges to that viewpoint, neuroscience hypothesises that consciousness is created by the interoperation of different parts of the brain, known as the neural correlates of consciousness or NCC. AC proponents claim it is possible to construct systems (such as computer systems) to mimic NCC interoperability [224]. Artificial consciousness concepts are also pondered in artificial intelligence philosophy through questions about mind, consciousness, and mental states [220].

Despite the recent successes in AI and deep learning, we are likely decades away from achieving any of them. AGI would be accomplished through a combination of deep learning and the existing tools like self-play and domain randomisation, or a new formulation of the same problem may be used. The questions, how will deep learning impact the development of AGI.

These characteristics have a moral component since a computer with this strong AI level may have rights similar to those of non-human animals. As a result, preliminary work on approaches to incorporating complete ethical agents with current legal and social systems has been done. The legal status and rights of strong AI have been the subject of these approaches [225].

The human mind possesses general intelligence, but most artificial intelligence does not. The most fundamental question in AGI ethics is whether to regard AGI as an academic endeavour or as something that has the potential to affect society and the entire world. The possibility of AGI disaster is based on the idea that AGI will one day outsmart humanity, seize control of the world, and achieve whatever objectives it is programmed to pursue. Catastrophe will ensue unless it is configured with aims that are secure for mankind, or something else one cares about [226].

Few questions are interesting to explore at this level: How can we evaluate the effectiveness of the current deep learning for achieving AGI dream? How can a cognitively automated reasoning framework complement everyday human reasoning? Do we need to change the deep learning strategy to achieve AGI fundamentally, and if yes, then How?

III. Future Directions

Overall, we have seen a remarkable success profile of deep learning. The emergence of big data and powerful computing has revolutionalised the field. The interest and attention of the deep learning research community, academia and industry sponsorship has contributed towards the developments of applied deep learning in business and industry. The last decade has seen the development of AlexNet [227], ResNet [27], GAN [228], Deep Q-learning [211], and Transformer networks [229]. There is excitement about new learning strategies like self-supervised learning [230], contrastive learning [231], and meta-learning [232]. The importance of unsupervised approach to deep learning is illustrated LeCun cake analogy.

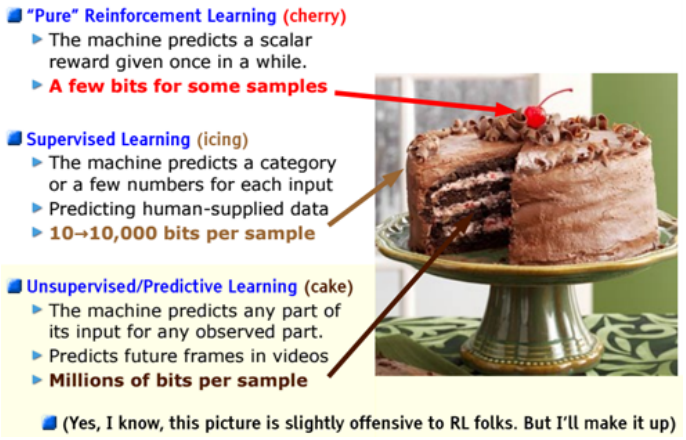

Fig. 31. Original LeCun cake analogy slide presented at NIPS 2016, the highlighted area has now been updated. Source [233]

IV. Conclusion

In this paper, we have provided a comprehensive overview of the progress in deep learning with a special focus on computer vision. We provided a classification of the most well-defined research problems in the field with progress to date, achievements and challenges. We tried to hight different research questions to excite new researchers in the area. We reviewed the various approaches in deep learning in the context of their improvement and highlighted the research gaps. We hope that this comprehensive review will summarise the deep problems in deep understanding for our research community and would serve as an exciting tool for further development in the field.

References

[1] Anwaar Ulhaq. Deep learning, past present and future: An odyssey. 2021.

[2] Nina Crummy. Six honest serving men: a basic methodology for the study of small finds. *Roman Finds: Context and Theory. Oxford: Oxbow Books*, pages 59–66, 2007.

[3] David Sharp et al. Kipling's guide to writing a scientific paper. *Croatian medical journal*, 43(3):262–267, 2002.

[4] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[5] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[6] Imagenet benchmark (image classification) — papers with code. https://paperswithcode.com/sota/image-classification-on-imagenet. (Accessed on 04/15/2021).

[7] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017.

[8] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[9] Mohammad Hossin and MN Sulaiman. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1, 2015.

[10] Feng Yan, Olatunji Ruwase, Yuxiong He, and Trishul Chilimbi. Performance modeling and scalability optimization of distributed deep learning systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1355–1364, 2015.

[11] Peter H Jin, Qiaochu Yuan, Forrest Iandola, and Kurt Keutzer. How to scale distributed deep learning? *arXiv preprint arXiv:1611.04581*, 2016.

[12] Jason Jinquan Dai, Yiheng Wang, Xin Qiu, Ding Ding, Yao Zhang, Yanzhang Wang, Xianyan Jia, Cherry Li Zhang, Yan Wan, Zhichao Li, et al. Bigdl: A distributed deep learning framework for big data. In *Proceedings of the ACM Symposium on Cloud Computing*, pages 50–60, 2019.

[13] Apache spark™ - unified analytics engine for big data. https://spark.apache.org/. (Accessed on 04/15/2021).

[14] Anders Arpteg, Björn Brinne, Luka Crnkovic-Friis, and Jan Bosch. Software engineering challenges of deep learning. In *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 50–59. IEEE, 2018.

[15] Amir Yazdanbakhsh, Jongse Park, Hardik Sharma, Pejman Lotfi-Kamran, and Hadi Esmaeilzadeh. Neural acceleration for gpu throughput processors. In *Proceedings of the 48th International Symposium on Microarchitecture*, pages 482–493, 2015.

[16] Juhyun Lee, Nikolay Chirkov, Ekaterina Ignasheva, Yury Pisarchyk, Mogan Shieh, Fabio Riccardi, Raman Sarokin, Andrei Kulik, and Matthias Grundmann. On-device neural net inference with mobile gpus. *arXiv preprint arXiv:1907.01989*, 2019.

[17] Ruben Mayer and Hans-Arno Jacobsen. Scalable deep learning on distributed infrastructures: Challenges, techniques, and tools. *ACM Computing Surveys (CSUR)*, 53(1):1–37, 2020.

[18] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. *arXiv preprint arXiv:1608.03665*, 2016.

[19] Yutao Huang, Xiaoqiang Ma, Xiaoyi Fan, Jiangchuan Liu, and Wei Gong. When deep learning meets edge computing. In *2017 IEEE 25th international conference on network protocols (ICNP)*, pages 1–2. IEEE, 2017.

[20] Leandro Parente, Evandro Taquary, Ana Paula Silva, Carlos Souza, and Laerte Ferreira. Next generation mapping: Combining deep learning, cloud computing, and big remote sensing data. *Remote Sensing*, 11(23):2881, 2019.

[21] Jiasi Chen and Xukan Ran. Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8):1655–1674, 2019.

[22] Waldemar Hummer, Vinod Muthusamy, Thomas Rausch, Parijat Dube, Kaoutar El Maghraoui, Anupama Murthi, and Punleuk Oum. Modelops: Cloud-based lifecycle management for reliable and trusted ai. In *2019 IEEE International Conference on Cloud Engineering (IC2E)*, pages 113–120. IEEE, 2019.

[23] Ekaba Bisong. *Building machine learning and deep learning models on Google cloud platform*. Springer, 2019.

[24] Ameet V Joshi. Amazon's machine learning toolkit: Sagemaker. In *Machine Learning and Artificial Intelligence*, pages 233–243. Springer, 2020.

[25] Jeff Barnes. Azure machine learning. *Microsoft Azure Essentials. 1st ed, Microsoft*, 2015.

[26] Omer Akgul, H Ibrahim Penekli, and Yakup Genc. Applying deep learning in augmented reality tracking. In *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 47–54. IEEE, 2016.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] Hiroaki Mikami, Hisahiro Suganuma, Yoshiki Tanaka, Yuichi Kageyama, et al. Massively distributed sgd: Imagenet/resnet-50 training in a flash. *arXiv preprint arXiv:1811.05233*, 2018.

[29] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*, 2017.

[30] James O' Neill. An overview of neural network compression. *arXiv preprint arXiv:2006.03669*, 2020.

[31] Helmut Bolcskei, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. Optimal approximation with sparsely connected deep neural networks. *SIAM Journal on Mathematics of Data Science*, 1(1):8–45, 2019.

[32] An Xu, Zhouyuan Huo, and Heng Huang. On the acceleration of deep learning model parallelism with staleness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2088–2097, 2020.

[33] Zhihao Jia, Matei Zaharia, and Alex Aiken. Beyond data and model parallelism for deep neural networks. *arXiv preprint arXiv:1807.05358*, 2018.

[34] Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016.

[35] Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3):1–18, 2017.

[36] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[37] Yue Cao, Mingsheng Long, Jianmin Wang, Han Zhu, and Qingfu Wen. Deep quantization network for efficient image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[38] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. Deep hashing network for efficient similarity retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[39] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[40] Sridhar Swaminathan, Deepak Garg, Rajkumar Kannan, and Frederic Andres. Sparse low rank factorization for deep neural network compression. *Neurocomputing*, 398:185–196, 2020.

[41] Genta Indra Winata, Andrea Madotto, Jamin Shin, Elham J Barezi, and Pascale Fung. On the effectiveness of low-rank matrix factorization for lstm model compression. *arXiv preprint arXiv:1908.09982*, 2019.

[42] Etienne Dupuis, David Novo, Ian O'Connor, and Alberto Bosio. On the automatic exploration of weight sharing for deep neural network compression. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1319–1322. IEEE, 2020.

[43] Shell Xu Hu, Sergey Zagoruyko, and Nikos Komodakis. Exploring weight symmetry in deep neural networks. *Computer Vision and Image Understanding*, 187:102786, 2019.

[44] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1365–1374, 2019.

[45] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017.

[46] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685. PMLR, 2019.

[47] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[48] Sarit Khirirat, Hamid Reza Feyzmahdavian, and Mikael Johansson. Mini-batch gradient descent: Faster convergence under data sparsity. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2880–2887. IEEE, 2017.

[49] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013.

[50] Timothy Dozat. Incorporating nesterov momentum into adam. 2016.

[51] Agnes Lydia and Sagayaraj Francis. Adagrad—an optimizer for stochastic gradient descent. *Int. J. Inf. Comput. Sci.*, 6(5), 2019.

[52] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[53] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[54] Jan N Van Rijn and Frank Hutter. Hyperparameter importance across datasets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2367–2376, 2018.

[55] Marc Claesen and Bart De Moor. Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*, 2015.

[56] Rabiya Khalid and Nadeem Javaid. A survey on hyperparameters optimization algorithms of forecasting models in smart grid. *Sustainable Cities and Society*, page 102275, 2020.

[57] Xin He, Kaiyong Zhao, and Xiaowen Chu. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, 2021.

[58] Ekaba Bisong. Google automl: cloud vision. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pages 581–598. Springer, 2019.

[59] Min Wu, Weihua Ma, Yue Li, and Xiongbo Zhao. Automatic optimization of super parameters based on model pruning and knowledge distillation. In *2020 International Conference on Computer Engineering and Intelligent Control (ICCEIC)*, pages 111–116. IEEE, 2020.

[60] Radwa Elshawi, Mohamed Maher, and Sherif Sakr. Automated machine learning: State-of-the-art and open challenges. *arXiv preprint arXiv:1906.02287*, 2019.

[61] Thomas Elsken, Jan Hendrik Metzen, Frank Hutter, et al. Neural architecture search: A survey. *J. Mach. Learn. Res.*, 20(55):1–21, 2019.

[62] Kaicheng Yu, Christian Sciuto, Martin Jaggi, Claudiu Musat, and Mathieu Salzmann. Evaluating the search phase of neural architecture search. *arXiv preprint arXiv:1902.08142*, 2019.

[63] Yesmina Jaafra, Jean Luc Laurent, Aline Deruyver, and Mohamed Saber Naceur. Reinforcement learning for neural architecture search: A review. *Image and Vision Computing*, 89:57–66, 2019.

[64] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

[65] Asaf Noy, Niv Nayman, Tal Ridnik, Nadav Zamir, Sivan Doveh, Itamar Friedman, Raja Giryes, and Lihi Zelnik. Asap: Architecture search, anneal and prune. In *International Conference on Artificial Intelligence and Statistics*, pages 493–503. PMLR, 2020.

[66] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *International Conference on Machine Learning*, pages 4095–4104. PMLR, 2018.

[67] Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu. Hierarchical representations for efficient architecture search. *arXiv preprint arXiv:1711.00436*, 2017.

[68] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789, 2019.

[69] Ruochen Wang, Minhao Cheng, Xiangning Chen, Xiaocheng Tang, and Cho-Jui Hsieh. Rethinking architecture selection in differ-entiable nas. In *International Conference on Learning Representations*, 2021.

[70] Haifeng Jin, Qingquan Song, and Xia Hu. Auto-keras: An efficient neural architecture search system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1946–1956, 2019.

[71] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. Snas: stochastic neural architecture search. *arXiv preprint arXiv:1812.09926*, 2018.

[72] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[73] Frank Hutter, Jörg Lücke, and Lars Schmidt-Thieme. Beyond manual tuning of hyperparameters. *KI-Künstliche Intelligenz*, 29(4):329–337, 2015.

[74] Rasmiranjan Mohakud and Rajashree Dash. Survey on hyperparameter optimization using nature-inspired algorithm of deep convolution neural network. In *Intelligent and Cloud Computing*, pages 737–744. Springer, 2021.

[75] Robert Geirhos, Carlos R Medina Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. *arXiv preprint arXiv:1808.08750*, 2018.

[76] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017.

[77] Will Koehrsen. Overfitting vs. underfitting: A complete example. *Towards Data Science*, 2018.

[78] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[79] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[80] Robert Moore and John DeNero. L1 and l2 regularization for multiclass hinge loss models. In *Symposium on machine learning in speech and language processing*, 2011.

[81] Aidan N Gomez, Ivan Zhang, Siddhartha Rao Kamalakara, Divyam Madaan, Kevin Swersky, Yarin Gal, and Geoffrey E Hinton. Learning sparse networks using targeted dropout. *arXiv preprint arXiv:1905.13678*, 2019.

[82] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[83] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[84] Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. Faster autoaugment: Learning augmentation strategies using backpropagation. In *European Conference on Computer Vision*, pages 1–16. Springer, 2020.

[85] Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. *arXiv preprint arXiv:1410.1141*, 2014.

[86] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[87] Ari S Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. *arXiv preprint arXiv:1803.06959*, 2018.

[88] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[89] Emilio Soria Olivas, Jos David Mart Guerrero, Marcelino Martinez-Sober, Jose Rafael Magdalena-Benedito, L Serrano, et al. *Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques: Algorithms, methods, and techniques*. IGI Global, 2009.

[90] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[91] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[92] Jingjing Li, Erpeng Chen, Zhengming Ding, Lei Zhu, Ke Lu, and Heng Tao Shen. Maximum density divergence for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[93] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[94] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4500–4509, 2018.

[95] The world's most valuable resource is no longer oil, but data — the economist. https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data. (Accessed on 04/16/2021).

[96] Howard Besser. Visual access to visual images: the uc berkeley image database project. 1990.

[97] Howard Besser. Image databases: The first decade, the present, and the future. *Digital Image Access & Retrieval [papers presented at the 1996 Clinic on Library Applications of Data Processing, March 24-26, 1996 Urbana-Champaign]*, 1997.

[98] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-100). 1996.

[99] Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 139–147, 2010.

[100] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[101] Josh Eno and Craig W Thompson. Generating synthetic data to match data mining patterns. *IEEE Internet Computing*, 12(3):78–82, 2008.

[102] Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. A review of feature selection methods on synthetic data. *Knowledge and information systems*, 34(3):483–519, 2013.

[103] Amlan Kar, Aayush Prakash, Ming-Yu Liu, Eric Cameracci, Justin Yuan, Matt Rusiniak, David Acuna, Antonio Torralba, and Sanja Fidler. Meta-sim: Learning to generate synthetic datasets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4551–4560, 2019.

[104] Daniel J Fagnant and Kara Kockelman. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, 77:167–181, 2015.

[105] Hesham M Eraqi, Mohamed N Moustafa, and Jens Honer. End-to-end deep learning for steering autonomous vehicles considering temporal dependencies. *arXiv preprint arXiv:1710.03804*, 2017.

[106] Branislav Kisačanin. Deep learning for autonomous vehicles. In *2017 IEEE 47th International Symposium on Multiple-Valued Logic (ISMVL)*, pages 142–142. IEEE, 2017.

[107] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8660–8669, 2019.

[108] Shaoshan Liu. *Engineering Autonomous Vehicles and Robots: The DragonFly Modular-based Approach*. John Wiley & Sons, 2020.

[109] Gina Neff. Talking to bots: Symbiotic agency and the case of tay. *International Journal of Communication*, 2016.

[110] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[111] Keith Kirkpatrick. Battling algorithmic bias: How do we ensure algorithms treat us fairly? *Communications of the ACM*, 59(10):16–17, 2016.

[112] Joseph P Robinson, Gennady Livitz, Yann Henon, Can Qin, Yun Fu, and Samson Timoner. Face recognition: too bias, or not too bias? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–1, 2020.

[113] James Heckman. Varieties of selection bias. *The American Economic Review*, 80(2):313–318, 1990.

[114] Bet Caeyers and Marcel Fafchamps. Exclusion bias in the estimation of peer effects. Technical report, National Bureau of Economic Research, 2016.

[115] Paul E Utgoff. *Machine learning of inductive bias*, volume 15. Springer Science & Business Media, 2012.

[116] Amina Adadi. A survey on data-efficient algorithms in big data era. *Journal of Big Data*, 8(1):1–54, 2021.

[117] Ričards Marcinkevičs and Julia E Vogt. Interpretability and explainability: A machine learning zoo mini-tour. *arXiv preprint arXiv:2012.01805*, 2020.

[118] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[119] Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. *Explainable AI: interpreting, explaining and visualizing deep learning*, volume 11700. Springer Nature, 2019.

[120] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[121] Philipp Schmidt and Felix Biessmann. Quantifying interpretability and trust in machine learning systems. *arXiv preprint arXiv:1901.08558*, 2019.

[122] Julia Ling, Maxwell Hutchinson, Erin Antono, Brian DeCost, Elizabeth A Holm, and Bryce Meredig. Building data-driven models with microstructural images: Generalization and interpretability. *Materials Discovery*, 10:19–28, 2017.

[123] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.

[124] Freddy Lecue. On the role of knowledge graphs in explainable ai. *Semantic Web*, 11(1):41–51, 2020.

[125] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016.

[126] Iam Palatnik De Sousa, Marley Maria Bernardes Rebuzzi Vellasco, and Eduardo Costa Da Silva. Local interpretable model-agnostic explanations for classification of lymph node metastases. *Sensors (Basel, Switzerland)*, 19(13), 2019.

[127] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017.

[128] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

[129] Sarah Tan, Rich Caruana, Giles Hooker, Paul Koch, and Albert Gordo. Learning global additive explanations for neural nets using model distillation. *arXiv preprint arXiv:1801.08640*, 2018.

[130] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

[131] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

[132] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. 2015.

[133] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[134] Jinkyu Kim and John Canny. Interpretable learning for self-driving cars by visualizing causal attention. In *Proceedings of the IEEE international conference on computer vision*, pages 2942–2950, 2017.

[135] Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. Metrics for explainable ai: Challenges and prospects. *arXiv preprint arXiv:1812.04608*, 2018.

[136] Samaneh Mahdavifar and Ali A Ghorbani. Application of deep learning to cybersecurity: A survey. *Neurocomputing*, 347:149–176, 2019.

[137] Jia Song and Jim Alves-Foss. The darpa cyber grand challenge: A competitor's perspective. *IEEE Security & Privacy*, 13(6):72–76, 2015.

[138] Chathurika S Wickramasinghe, Daniel L Marino, Kasun Amarasinghe, and Milos Manic. Generalization of deep learning for cyber-physical system security: A survey. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, pages 745–751. IEEE, 2018.

[139] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[140] Guofu Li, Pengjia Zhu, Jin Li, Zhemin Yang, Ning Cao, and Zhiyi Chen. Security matters: A survey on adversarial machine learning. *arXiv preprint arXiv:1810.07339*, 2018.

[141] Ho Bae, Jaehee Jang, Dahuin Jung, Hyemi Jang, Heonseok Ha, and Sungroh Yoon. Security and privacy issues in deep learning. *arXiv preprint arXiv:1807.11655*, 2018.

[142] Rohan Taori, Amog Kamsetty, Brenton Chu, and Nikita Vemuri. Targeted adversarial examples for black box audio systems. In *2019 IEEE Security and Privacy Workshops (SPW)*, pages 15–20. IEEE, 2019.

[143] Xianmin Wang, Jing Li, Xiaohui Kuang, Yu-an Tan, and Jin Li. The security of machine learning in an adversarial setting: A survey. *Journal of Parallel and Distributed Computing*, 130:12–23, 2019.

[144] Hamza Fawzi, Paulo Tabuada, and Suhas Diggavi. Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Transactions on Automatic control*, 59(6):1454–1467, 2014.

[145] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[146] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

[147] Daniel Jakubovitz and Raja Giryes. Improving dnn robustness to adversarial attacks using jacobian regularization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 514–529, 2018.

[148] Fatemehsadat Mirshghallah, Mohammadkazem Taram, Praneeth Vepakomma, Abhishek Singh, Ramesh Raskar, and Hadi Esmaeilzadeh. Privacy in deep learning: A survey. *arXiv preprint arXiv:2004.12254*, 2020.

[149] Ximeng Liu, Lehui Xie, Yaopeng Wang, Jian Zou, Jinbo Xiong, Zuobin Ying, and Athanasios V Vasilakos. Privacy and security issues in deep learning: A survey. *IEEE Access*, 2020.

[150] Xueluan Gong, Qian Wang, Yanjiao Chen, Wang Yang, and Xinchang Jiang. Model extraction attacks and defenses on cloud-based machine learning models. *IEEE Communications Magazine*, 58(12):83–89, 2020.

[151] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[152] Ann Cavoukian et al. Privacy by design: The 7 foundational principles. *Information and privacy commissioner of Ontario, Canada*, 5:12, 2009.

[153] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10:3152676, 2017.

[154] Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015.

[155] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[156] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

[157] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, and Bingsheng He. A survey on federated learning systems: vision, hype and reality for data privacy and protection. *arXiv preprint arXiv:1907.09693*, 2019.

[158] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

[159] Anwaar Ulhaq and Oliver Burmeister. Covid-19 imaging data privacy by federated learning design: A theoretical framework. *arXiv preprint arXiv:2010.06177*, 2020.

[160] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.

[161] Mengnan Du, Fan Yang, Na Zou, and Xia Hu. Fairness in deep learning: A computational perspective. *IEEE Intelligent Systems*, 2020.

[162] Catherina Xu, Christina Greer, Manasi N Joshi, and Tulsee Doshi. Fairness indicators demo: Scalable infrastructure for fair ml systems. 2020.

[163] Qianwen Wang, Zhenhua Xu, Zhutian Chen, Yong Wang, Shixia Liu, and Huamin Qu. Visual analysis of discrimination in machine learning. *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[164] Stefan Feuerriegel, Mateusz Dolata, and Gerhard Schwabe. Fair ai: Challenges and opportunities. *Business & information systems engineering*, 62:379–384, 2020.

[165] Keng Siau and Weiyu Wang. Building trust in artificial intelligence, machine learning, and robotics. *Cutter Business Technology Journal*, 31(2):47–53, 2018.

[166] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

[167] Nina Narodytska, Shiva Kasiviswanathan, Leonid Ryzhyk, Mooly Sagiv, and Toby Walsh. Verifying properties of binarized deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[168] Lindsey Kuper, Guy Katz, Justin Gottschlich, Kyle Julian, Clark Barrett, and Mykel Kochenderfer. Toward scalable verification for safety-critical deep networks. *arXiv preprint arXiv:1801.05950*, 2018.

[169] Asim Abdulkhaleq, Stefan Wagner, Daniel Lammering, Hagen Boehmert, and Pierre Blueher. Using stpa in compliance with iso 26262 for developing a safe architecture for fully automated vehicles. *arXiv preprint arXiv:1703.03657*, 2017.

[170] Chen Yan, Wenyuan Xu, and Jianhao Liu. Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. *Def Con*, 24(8):109, 2016.

[171] Thomas J Hwang, Aaron S Kesselheim, and Kerstin N Vokinger. Lifecycle regulation of artificial intelligence–and machine learning–based software devices in medicine. *Jama*, 322(23):2285–2286, 2019.

[172] Samuele Lo Piano. Ethical principles in machine learning and artificial intelligence: cases from the field and possible ways forward. *Humanities and Social Sciences Communications*, 7(1):1–7, 2020.

[173] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[174] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, 64:131–148, 2020.

[175] Jan Kietzmann, Linda W Lee, Ian P McCarthy, and Tim C Kietzmann. Deepfakes: Trick or treat? *Business Horizons*, 63(2):135–146, 2020.

[176] Ashish Jaiman. Debating the ethics of deepfakes. *in a Pandemic World*, page 75.

[177] Thanh Thi Nguyen, Cuong M Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, and Saeid Nahavandi. Deep learning for deepfakes creation and detection: A survey. *arXiv preprint arXiv:1909.11573*, 2019.

[178] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854*, 2019.

[179] Steven Umbrello and Angelo Frank De Bellis. A value-sensitive design approach to intelligent agents. *Artificial Intelligence Safety and Security (2018) CRC Press (. ed) Roman Yampolskiy*, 2018.

[180] Robert J Sternberg. *Handbook of creativity*. Cambridge University Press, 1999.

[181] Mark A Runco and Garrett J Jaeger. The standard definition of creativity. *Creativity research journal*, 24(1):92–96, 2012.

[182] Margaret A Boden. Creativity. In *Artificial intelligence*, pages 267–291. Elsevier, 1996.

[183] Caterina Moruzzi. Measuring creativity: an account of natural and artificial creativity. *European Journal for Philosophy of Science*, 11(1):1–20, 2021.

[184] Massimiliano Palmiero, Raffaella Nori, Vincenzo Aloisi, Martina Ferrara, and Laura Piccardi. Domain-specificity of creativity: A study on the relationship between visual creativity and visual mental imagery. *Frontiers in Psychology*, 6:1870, 2015.

[185] Paul Locher. How does a visual artist create an artwork. *The Cambridge handbook of creativity*, pages 131–144, 2010.

[186] Wenfu Li, Junyi Yang, Qinglin Zhang, Gongying Li, and Jiang Qiu. The association between resting functional connectivity and visual creativity. *Scientific reports*, 6(1):1–10, 2016.

[187] Nicola De Pisapia, Francesca Bacci, Danielle Parrott, and David Melcher. Brain networks for visual creativity: a functional connectivity study of planning a visual artwork. *Scientific reports*, 6(1):1–11, 2016.

[188] Melissa Ellamil, Charles Dobson, Mark Beeman, and Kalina Christoff. Evaluative and generative modes of thought during the creative process. *Neuroimage*, 59(2):1783–1794, 2012.

[189] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Innovation engines: Automated creativity and improved stochastic optimization via deep learning. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 959–966, 2015.

[190] John H Flowers and Calvin P Garbin. Creativity and perception. In *Handbook of creativity*, pages 147–162. Springer, 1989.

[191] Simona Cohen. Some aspects of michelangelo's creative process. *Artibus et Historiae*, pages 43–63, 1998.

[192] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[193] Aäron van den Oord and Nal Kalchbrenner. Pixel rnn. 2016.

[194] Marcus du Sautoy. Can ai ever be truly creative? *New Scientist*, 242(3229):38–41, 2019.

[195] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[196] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[197] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[198] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, M Chen, R Child, V Misra, P Mishkin, G Krueger, S Agarwal, et al. Dall· e: Creating images from text. *OpenAI Blog*, 2021.

[199] Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020.

[200] Philippe Pasquier, Arne Eigenfeldt, Oliver Bown, and Shlomo Dubnov. An introduction to musical metacreation. *Computers in Entertainment (CIE)*, 14(2):1–14, 2017.

[201] Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):187–203, 2019.

[202] Fabio Zünd, Mattia Ryffel, Stéphane Magnenat, Alessia Marra, Maurizio Nitti, Mubbasir Kapadia, Gioacchino Noris, Kenny Mitchell, Markus Gross, and Robert W Sumner. Augmented creativity: Bridging the real and virtual worlds to enhance creative play. In *SIGGRAPH Asia 2015 Mobile Graphics and Interactive Applications*, pages 1–7. 2015.

[203] Jacqueline Fendt and Renata Kaminska-Labbé. Relevance and creativity through design-driven action research: Introducing pragmatic adequacy. *European Management Journal*, 29(3):217–233, 2011.

[204] Ben Goertzel. Artificial general intelligence: concept, state of the art, and future prospects. *Journal of Artificial General Intelligence*, 5(1):1, 2014.

[205] Allen Newell and Herbert A Simon. Computer science as empirical inquiry: Symbols and search. In *ACM Turing award lectures*, page 1975. 2007.

[206] Roman V Yampolskiy. Ai-complete, ai-hard, or ai-easy–classification of problems in ai. In *The 23rd Midwest Artificial Intelligence and Cognitive Science Conference, Cincinnati, OH, USA*, 2012.

[207] Roman V Yampolskiy. Turing test as a defining feature of ai-completeness. In *Artificial intelligence, evolutionary computing and metaheuristics*, pages 3–17. Springer, 2013.

[208] Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, 2019.

[209] Alex Alaniz. Early doctrine for and a path to warfare by artificial general intelligence (agi). 2018.

[210] Sean D Holcomb, William K Porter, Shaun V Ault, Guifen Mao, and Jin Wang. Overview on deepmind and its alphago zero ai. In *Proceedings of the 2018 international conference on big data and education*, pages 67–71, 2018.

[211] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[212] Rasool Fakoor, Pratik Chaudhari, Stefano Soatto, and Alexander J Smola. Meta-q-learning. *arXiv preprint arXiv:1910.00125*, 2019.

[213] Niket Tandon, Aparna S Varde, and Gerard de Melo. Commonsense knowledge in machine intelligence. *ACM SIGMOD Record*, 46(4):49–52, 2018.

[214] Henry Markram. The human brain project. *Scientific American*, 306(6):50–55, 2012.

[215] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[216] Tianhong Dai, Kai Arulkumaran, Tamara Gerbert, Samyakh Tukra, Feryal Behbahani, and Anil Anthony Bharath. Analysing deep reinforcement learning agents trained with domain randomisation. *arXiv preprint arXiv:1912.08324*, 2019.

[217] Joseph Suarez, Yilun Du, Phillip Isola, and Igor Mordatch. Neural mmo: A massively multiagent game environment for training and evaluating intelligent agents. *arXiv preprint arXiv:1903.00784*, 2019.

[218] Nils J Nilsson. Human-level artificial intelligence? be serious! *AI magazine*, 26(4):68–68, 2005.

[219] Antonio Chella and Riccardo Manzotti. *Artificial consciousness*. Andrews UK Limited, 2013.

[220] Giorgio Buttazzo. Artificial consciousness: Utopia or real possibility? *Computer*, 34(7):24–30, 2001.

[221] Alain Cardon. Artificial consciousness, artificial emotions, and autonomous robots. *Cognitive processing*, 7(4):245–267, 2006.

[222] David Gamez. Progress in machine consciousness. *Consciousness and cognition*, 17(3):887–910, 2008.

[223] Igor Aleksander. Artificial neuroconsciousness an update. In *International Workshop on Artificial Neural Networks*, pages 566–583. Springer, 1995.

[224] David Gamez. The potential for consciousness of artificial systems. *International Journal of Machine Consciousness*, 1(02):213–223, 2009.

[225] Steven Livingston and Mathias Risse. The future impact of artificial intelligence on humans and human rights. *Ethics & international affairs*, 33(2):141–158, 2019.

[226] David Kelley and Kyrtin Atreides. Agi protocol for the ethical treatment of artificial general intelligence systems. *Procedia Computer Science*, 169:501–506, 2020.

[227] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[228] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[229] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[230] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[231] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[232] Fengwei Zhou, Bin Wu, and Zhenguo Li. Deep meta-learning: Learning to learn in the concept space. *arXiv preprint arXiv:1802.03596*, 2018.

[233] Yann LeCun. Nips 2016 schedule. https://nips.cc/Conferences/2016/Schedule?showEvent=6197, 2016. (Accessed on 04/15/2021).