

CNN Based Disease Prevention

Sumanth Kota *

*(Electronics and Instrumentation Engineering,
Keshav Memorial Institute of Technology,
Jawaharlal Technological University, Hyderabad
Email: Sumanth.kota111999@gmail.com)

Abstract

Ever since 2019 life of the world has become pathetic due to the prevalence of Coronavirus. It has been into records as one of the most dangerous death waves, one has ever occurred to mankind. Scientists and Bio-engineers have been working on the eradication of this virus. Vaccination has been done for a few days all around the globe, but somehow it turns to be ineffective to prevent the disease. My technique can help to find an appropriate vaccine, it can help in eliminating the virus by different scientific procedures. This research article is solely based on *Microbiology* and *Artificial Intelligence* techniques. It uses the concepts of microbiology to train the AI model efficiently with loads of data, that are already available in abundance in the microbiology labs, derived by the powerful electronic microscopes. Using *Convolutional Neural Networks*, by giving highly dense *Deep Neural Network* we iterate through the provided data, therefore implemented by *Deep Learning* techniques. This would also need the data of already existing enzymes. This whole procedure resembles gene therapy or DNA manipulation, merely.

Keywords—*Coronavirus, Bio-engineers, Eradication, Vaccine, Microbiology, Artificial Intelligence, Convolutional Neural Networks, Deep Learning, Deep Neural Networks, Enzymes, DNA, Gene Therapy.*

I. INTRODUCTION

This involves a few crucial steps to finish the whole procedure. That requires expertise in multiple fields of biology like, microbiology, genetical engineering, medical sciences, biomedical engineering; simultaneously it needs proficiency in Computer Vision, Machine Learning, Deep learning, Neural networks, Artificial Intelligence, and python.

This experimental project would need large amounts of Image data to be processed, and trained to the model. The AI model will then be able to recognize different categories we train to it. That categories include different kinds of viruses, chemical bonds, and enzymes.

The training process is procedural. It must be executed in chronological order. The model must be trained first with the microscopic image samples of

different kinds of viruses, followed by different types of chemical bonds different viruses have, based on the result of the first step the model must be able to list the chemical bonds that a specific virus contains. Further, the process would be to train with the available enzyme data, this is trained in such a way that the model should be able to pick a particular enzyme based on the chemical bonds. Like, a specific chemical bond takes a particular enzyme, we must make sure to provide proper data to the AI model to work effectively.

Microscope image processing is a broad term that covers Digital Image Processing techniques to process, analyze and present images obtained from a microscope.

Such processing is now commonplace in a number of fields such

as medicine, biological research, cancer research, drug testing, metallurgy, etc.

Several manufacturers of microscopes now specifically design features that allow the microscopes to interface to an image processing system.

This whole process means Image Acquisition and Image Processing. The images will be acquired by using microscopes in laboratories, thus obtained data would be processed using Image Processing techniques, with the usage of Computer Vision.

Today, the acquisition is usually done using a CCD camera mounted in the optical path of the microscope. The camera may be full color or monochrome. Very often, very high-resolution cameras are employed to gain as much direct information as possible. Cryogenic cooling is also common, to minimize noise. Often digital cameras used for this application provide pixel intensity data to a resolution of 12-16 bits, much higher than is used in consumer imaging products.

Ironically, in recent years, much effort has been put into acquiring data at video rates, or higher (25-30 frames per second or higher). What was once easy with off-the-shelf video cameras now requires special, high-speed electronics to handle the vast digital data bandwidth.

Higher speed acquisition allows dynamic processes to be observed in real-time, or stored for later playback and analysis. Combined with the high image resolution, this approach can generate vast quantities of raw data, which can be a challenge to deal with, even with a modern computer system.

It should be observed that while current CCD detectors allow very high image resolution, often this involves a trade-off because, for a given chip size, as the pixel count increases, the pixel size decreases. As the pixels get smaller, their good depth decreases, reducing the number of electrons that can be stored. In turn, this results in a poorer signal-to-noise ratio.

For best results, one must select an appropriate sensor for a given application. Because microscope images have an intrinsic limiting resolution, it often makes little sense to use a noisy, high-resolution

detector for image acquisition. A more modest detector, with larger pixels, can often produce much higher quality images because of reduced noise. This is especially important in low-light applications such as fluorescence microscopy.

Moreover, one must also consider the temporal resolution requirements of the application. A lower resolution detector will often have a significantly higher acquisition rate, permitting the observation of faster events. Conversely, if the observed object is motionless, one may wish to acquire images at the highest possible spatial resolution without regard to the time required to acquire a single image.

2D Image Techniques

Image processing for microscopy application begins with fundamental techniques intended to most accurately reproduce the information contained in the microscopic sample. This might include adjusting the brightness and contrast of the image, averaging images to reduce image noise, and correcting for illumination non-uniformities. Such processing involves only basic arithmetic operations between images (i.e. addition, subtraction, multiplication, and division). The vast majority of processing done on microscope images is of this nature.

Another class of common 2D operations called image convolution are often used to reduce or enhance image details. Such "blurring" and "sharpening" algorithms in most programs work by altering a pixel's value based on a weighted sum of that and the surrounding pixels (a more detailed description of kernel-based convolution deserves an entry for itself) or by altering the frequency domain function of the image using Fourier Transform. Most image processing techniques are performed in the frequency domain.

Other basic two-dimensional techniques include operations such as image rotation, warping, color balancing, etc.

At times, advanced techniques are employed with the goal of "undoing" the distortion of the optical path of the microscope, thus eliminating distortions and blurring caused by the instrumentation. This process is called deconvolution, and a variety

of algorithms have been developed, some of the great mathematical complexity. The result is an image far sharper and clearer than could be obtained in the optical domain alone. This is typically a 3-dimensional operation, that analyzes a volumetric image (i.e. images taken at a variety of focal planes through the sample) and uses this data to reconstruct a more accurate 3-dimensional image.

3D Image Techniques

Another common requirement is to take a series of images at a fixed position but at different focal depths. Since most microscopic samples are essentially transparent, and the depth of field of the focused sample is exceptionally narrow, it is possible to capture images "through" a three-dimensional object using 2D equipment like confocal microscopes. Software is then able to reconstruct a 3D model of the original sample which may be manipulated appropriately. The processing turns a 2D instrument into a 3D instrument, which would not otherwise exist. In recent times this technique has led to many scientific discoveries in cell biology.

Analysis

Analysis of images will vary considerably according to application. Typical analysis includes determining where the edges of an object are, counting similar objects, calculating the area, perimeter length, and other useful measurements of each object. A common approach is to create an image mask that only includes pixels that match certain criteria, then perform simpler scanning operations on the resulting mask. It is also possible to label objects and track their motion over a series of frames in a video sequence.

Digital image processing is the use of a digital computer to process digital images through an algorithm. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems. The generation and development of digital image

processing are mainly affected by three factors: first, the development of computers; second, the development of mathematics (especially the creation and improvement of discrete mathematics theory);

Digital image processing allows the use of much more complex algorithms, and hence, can offer both more sophisticated performance at simple tasks, and the implementation of methods that would be impossible by analog means.

In particular, digital image processing is a concrete application of, and a practical technology based on:

- Classification
- Feature extraction
- Multi-scale signal analysis
- Pattern recognition
- Projection

Some techniques which are used in digital image processing include:

- Anisotropic diffusion
- Hidden Markov models
- Image editing
- Image restoration
- Independent component analysis
- Linear filtering
- Neural networks
- Partial differential equations
- Pixelation
- Point feature matching
- Principal components analysis
- Self-organizing maps
- Wavelets

Filtering

Digital filters are used to blur and sharpen digital images. Filtering can be performed by:

- convolution with kernels (filter array) in the spatial domain.

- masking specific frequency regions in the frequency (Fourier) domain.

3 × 3 Mean Blur

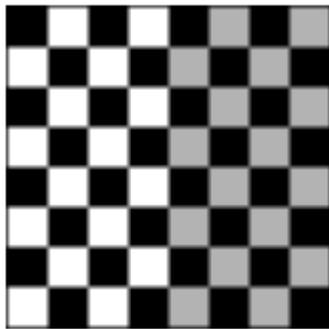


Figure-1: 3*3 Mean Blur Spatial Low-pass filter

(Digital_image_processing#/media/File:Affine_Transformation_Original_Checkerboard.jpg)

FFT Representation

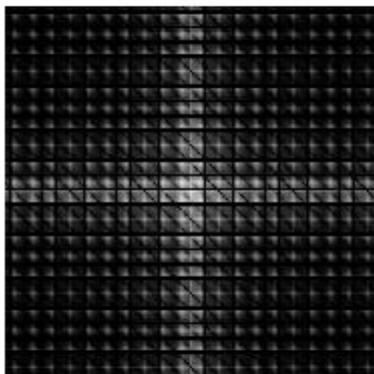


Figure- 2: Fourier Transform Representation
(Digital_image_processing#/media/File:Fourier_Space_Checkerboard.png)

Transformation Name	Affine Matrix	Example
Identity		
Reflection		
Scale		
Rotate		 where $\theta = \pi/6 = 30^\circ$

Figure-3: Affine transformations

Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.

Deep-learning architectures such as deep neural networks, deep belief networks, deep reinforcement learning, recurrent neural networks, and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, material inspection, and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.

Artificial neural networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have differences from biological brains. Specifically, neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analog.

The adjective "deep" in deep learning refers to the use of multiple layers in the network. Early work showed that a linear perceptron cannot be a universal classifier, but that a network with a nonpolynomial activation function with one hidden layer of unbounded width can. Deep learning is a modern variation that is concerned with an unbounded number of layers of bounded size, which permits practical application and optimized implementation while retaining theoretical universality under mild conditions. In deep learning the layers are also permitted to be heterogeneous and to deviate widely from biologically informed connectionist models, for the sake of efficiency, trainability, and understandability, whence the "structured" part.

Interpretations

Deep neural networks are generally interpreted in terms of the universal approximation theorem or probabilistic inference.

The classic universal approximation theorem concerns the capacity of feedforward neural networks with a single hidden layer of finite size to approximate continuous functions. In 1989, the first proof was published for sigmoid activation functions and was generalized to feed-forward multi-layer architectures in 1991 by Kurt Hornik. Recent work also showed that universal approximation also holds for non-bounded activation functions such as the rectified linear unit.

The universal approximation theorem for deep neural networks concerns the capacity of networks with bounded width but the depth is allowed to grow. Lu et al.^[23] proved that if the width of a deep neural network with ReLU activation is strictly larger than the input dimension, then the network can approximate any Lebesgue integrable function; If the width is smaller or equal to the input dimension, then the deep neural network is not a universal approximator.

The probabilistic interpretation derives from the field of machine learning. It features inference,^[29] as well as the optimization concepts of training and testing, related to fitting and generalization, respectively. More specifically, the probabilistic interpretation considers the activation nonlinearity as a cumulative distribution function. The probabilistic interpretation led to the introduction of dropout as a regularizer in neural networks. The probabilistic interpretation was introduced by researchers.

Artificial Neural Networks

Artificial neural networks (ANNs) or connectionist systems are computing systems inspired by the biological neural networks that constitute animal brains. Such systems learn (progressively improve their ability) to do tasks by considering examples, generally without task-specific programming. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the analytic results to identify cats in other images. They have found most use in applications difficult to express with a traditional computer algorithm using rule-based programming.

An ANN is based on a collection of connected units called artificial neurons, (analogous to biological neurons in a biological brain). Each connection (synapse) between neurons can transmit a signal to another neuron. The receiving (postsynaptic) neuron can process the signal(s) and then signal downstream neurons connected to it. Neurons may have states, generally represented by real numbers, typically between 0 and 1. Neurons and synapses may also have a weight that varies as learning proceeds, which can increase or decrease the strength of the signal that it sends downstream.

Typically, neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first (input) to the last (output) layer, possibly after traversing the layers multiple times.

The original goal of the neural network approach was to solve problems in the same way that a human brain would. Over time, attention focused on matching specific mental abilities, leading to deviations from biology such as backpropagation, or passing information in the reverse direction and adjusting the network to reflect that information.

Deep Neural Networks

A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers. There are different types of neural networks but they always consist of the same components: neurons, synapses, weights, biases, and functions.^[112] These components functioning similarly to the human brain and can be trained like any other ML algorithm.

For example, a DNN that is trained to recognize dog breeds will go over the given image and calculate the probability that the dog in the image is a certain breed. The user can review the results and select which probabilities the network should display (above a certain threshold, etc.), and return the proposed label. Each mathematical manipulation as such is considered a layer, and complex DNN has many layers, hence the name "deep" networks.

DNNs can model complex non-linear relationships. DNN architectures generate compositional models where the object is expressed as a layered

composition of primitives. The extra layers enable the composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network. For instance, it was proved that sparse multivariate polynomials are exponentially easier to approximate with DNNs than with shallow networks.

Deep architectures include many variants of a few basic approaches. Each architecture has found success in specific domains. It is not always possible to compare the performance of multiple architectures unless they have been evaluated on the same data sets.

DNNs are typically feedforward networks in which data flows from the input layer to the output layer without looping back. At first, the DNN creates a map of virtual neurons and assigns random numerical values, or "weights", to connections between them. The weights and inputs are multiplied and return an output between 0 and 1. If the network did not accurately recognize a particular pattern, an algorithm would adjust the weights.^[115] That way the algorithm can make certain parameters more influential until it determines the correct mathematical manipulation to fully process the data.

Recurrent neural networks (RNNs), in which data can flow in any direction, are used for applications such as language modeling. Long short-term memory is particularly effective for this use.

Convolutional deep neural networks (CNNs) are used in computer vision. CNNs also have been applied to acoustic modeling for automatic speech recognition (ASR).

Image Classification

It creates an image classifier using `keras.Sequential` model, and loads data using `preprocessing.image_dataset_from_directory`. You will gain practical experience with the following concepts:

- Efficiently loading a dataset off disk.

- Identifying overfitting and applying techniques to mitigate it, including data augmentation and Dropout.

This tutorial follows a basic machine learning workflow:

1. Examine and understand data
2. Build an input pipeline
3. Build the model
4. Train the model
5. Test the model
6. Improve the model and repeat the process

Image classification is the primary domain, in which deep neural networks play the most important role in medical image analysis. The image classification accepts the given input images and produces output classification for identifying whether the disease is present or not. E. Kim et al, proposed a CNN method that outperforms perfect image classification accuracy in cytopathology. Inception v3 architecture is one of the best methods for medical data analysis and has accomplished proficient human performance. The CNN architecture proposed by E. Hosseini-AsL et al.] uses three-dimensional convolutions to classify Alzheimer's disease. J. Kawahara et al. proposed a CNN-like architecture used for predicting the development of the brain. In image classification, CNNs are the recent state-of-the-art methods. The CNNs learned about natural images, showing strong performance and encountering the accuracy of human expert systems. Finally, these statements conclude that CNNs can be improved to control the essential architecture of medical images.

Image classification is a complex process that may be affected by many factors. Because classification results are the basis for many environmental and socioeconomic applications, scientists and practitioners have made great efforts in developing advanced classification approaches and techniques for improving classification accuracy. Image

classification is used a lot in basic fields like medicine, education, and security. Correct classification has vital importance, especially in medicine. Therefore, improved methods are needed in this field. The proposed deep CNNs are an often-used architecture for deep learning and have been widely used in computer vision and audio recognition. In the literature, different values of factors used for the CNNs are considered. From the results of the experiments on the CIFAR dataset, we argue that the network depth is the priority for improving the accuracy. It can not only improve the accuracy but also achieve the same high accuracy with less complexity compared to increasing the network width.

To classify a set of data into different classes or categories, the relationship between the data and the classes into which they are classified must be well understood. Generally, classification is done by a computer, so, to achieve classification by a computer, the computer must be trained. Sometimes it never gets sufficient accuracy with the results obtained, so training is a key to the success of classification. To improve the classification accuracy, inspired by the ImageNet challenge, the proposed work considers the classification of multiple images into the different categories (classes) with more accuracy in classification, cost reduction, and in a shorter time by applying parallelism using a deep neural network model.

The image classification problem requires determining the category (class) that an image belongs to. The problem is considerably complicated by the growth of categories' count, if several objects of different classes are present in the image and if the semantic class hierarchy is of interest because an image can belong to several categories simultaneously. Fuzzy classes present another difficulty for probabilistic categories' assignments. Moreover, a combination of different classification approaches is helpful for the improvement of classification accuracy.

Deep convolutional neural networks provide better results than existing methods in the literature due to advantages such as processing by extracting hidden features, allowing parallel processing, and real-time operation. The concept of convolutions in the context of neural networks begins with the idea of

layers consisting of neurons with a local receptive field, i.e., neurons which connect to a limited region of the input data and not the whole.

In this examination, the image classification process is performed by using TensorFlow, which is an open-source programming library in Python to manufacture our DCNN. We have considered the CIFAR-10 dataset, which contains 60,000 pictures. In the examination, 6000 and 3000 bits of data were taken from the related images for planning and testing exclusively the cat and pooch pictures taken from the CIFAR-10 dataset, which were resized, and histogram equalization operations were performed.

The experimental strategy was executed in Python using Ubuntu. We aimed for the best result in the image handling field. For implementing the operational parallelism, we had some limitations related to the CPU, which is useful only for sequential operations. But when it comes to parallel programming, GPUs are more prominent with CUDA. In the past, GPUs have been working as graphic accelerators. However, parallel programming has been developed as a powerful, general-purpose, and fully programmable parallel data processing approach for operations that require it.

CUDA is NVIDIA's parallel computing architecture that enables dramatic increases in computing performance by harnessing the power of the GPU (graphics processing unit). We perform the proposed method on Ubuntu 16.04 operating system using an NVIDIA Geforce GTX 680 with 2 GB of memory. In the developed model, GPU technology has been utilized while training.

Medical image classification

One of the most imperative problems faced in the domain area of image recognition is the classification of medical images. The major intention of medical image classification is to classify medical images into several elements to assist medical practitioners or physicists in diagnosing disease. Hence, medical image classification is split into two steps. The first and foremost step of medical image classification is to extract the essential features from the acquired input image. The second step in medical image classification is utilizing the features to construct

models that classify the image data set. In the recent past, medical practitioners customarily utilized their specialized experience to extract features so that classification of medical images could be performed into several classes. However, this manual medical image classification was found to be highly cumbersome and time-consuming.

Medical image classification involves the process of segregating medical-related information into a useful form. Classification of medical images is based on placing image pixels with similar values into groups. With the placement of similar values into groups, common pixels are identified and are denoted by these pixels. Hence, a correctly classified image usually denotes the areas on the ground that share specific features as specified in the classification scheme. From the above specification, images are classified into two types:

1. supervised classification and
2. unsupervised classification

a) Supervised classification

Supervised classification utilizes the spectral signatures acquired from training samples to classify a given input image.

b) Unsupervised classification

In contrast, an unsupervised classification identifies spectral classes present in a multiband image with human interference.

Based on image classification, a series of more complex tasks can be extended according to different application scenarios, such as object detection, object localization, image segmentation, etc. Object detection is a practical and challenging computer vision task, which can be regarded as a combination of image classification and localization. Given an image, the object detection system should be able to identify the objects in the image and provide their location.

As shown in Fig. 6.4, an object detection system can output several rectangular boxes and labels. Each rectangular box represents the boundary of the predicted object, as well as its category and position information. Both of these outputs are needed to be evaluated by developers. To evaluate the accuracy of the predicted boundary, the Intersection over Union (IoU) metric is introduced. To evaluate the correctness of the predicted category labels, the

Mean Average Precision (mAP) metric is introduced.

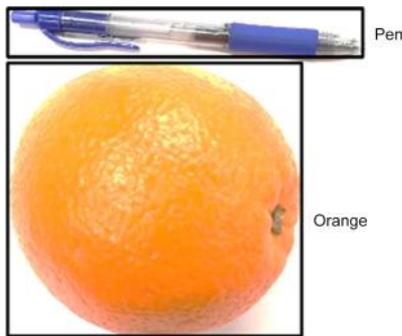


Figure-4: Object Detection

<https://ars.els-cdn.com/content/image/3-s2.0-B9780128234884000060-f06-04-9780128234884.jpg>

The concept of IoU is very intuitive; it illustrates the intersection of the predicted boundary and the ground-truth boundary. The bigger the IoU is, the higher is the performance of the prediction. If both intersections overlap entirely, the result is perfect.

$$\text{IoU} = \frac{\text{Predicted boundary} \cap \text{Ground-truth boundary}}{\text{Predicted boundary} \cup \text{Ground-truth boundary}}$$

In Fig. 6.5, a solid border indicates the ground-truth boundary of the object “orange” and a dashed border indicates the predicted boundary. In general, a matrix can be defined with the coordinates of the upper left corner and the lower right corner of the matrix, namely:

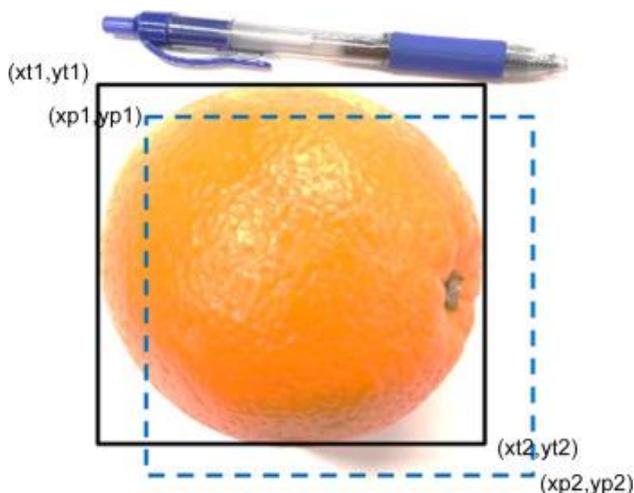


Figure-5: Truth Boundary

No matter for image classification or object detection or segmentation, feature extraction is the key to achieve a higher classification or detection performance. Most of the existing most features representations algorithms are based on handcrafted features, which mainly rely on experience and professional knowledge. Recently, the original deep learning, especially the CNN model, has attracted a lot of attention in the field of pattern recognition and computer vision. Although CNN models can automatically learn high-level semantic information from labeled data, CNNs require a fixed-size image as the input to model in previous approaches. To solve the problem of fixed size, two common operations are introduced, including crop operation or warp operation, which will affect final classification performance due to the loss of some discriminative spatial information. In this chapter, to eliminate the limitation of input images size, a novel CNN model framework with SPP-net is proposed to detect the fingerprint liveness. In addition, high-level structural features based on multiscale images are learned, so our method has the scale invariance and is more robust to scale-invariant. Weights and bias parameters learned in the pre-trained network model using ImageNet 2012 database are migrated to our model, which is viewed as the parameter initialization of our model architecture. Then, we fine-tune our SPP-net by employing the training samples from LivDet 2011 and LivDet 2013 databases. Our experiment proves that the classification performance of our model is better than other methods and is more suitable for FLD; it can better prevent spoofing attacks by artificial replicas.

Architecture

For Viruses:

- **Data Collection**

All the microscopic images of different kinds of viruses need to be collected, along with the images of the different bonds in the RNA of the viruses

Any virus is made up of RNA as their base structure

As the human DNA is made out of various organic compounds like C, H, O, N, all of them are in some order forms a certain strand, to form so they need to

form a bond with each other, that makes up a molecule.

Data Classification

The collected data needs to be classified.

We need to collect thousands of images of each kind of virus.

Each category needs to be put at a place with thousands of images, different perspectives.

After classifying and putting them in folders, we need to prepare a dataset on that.

Each kind of virus takes each label (label can be either numerical or a text)

For Enzymes:

- **Data Collection**

All the data of different kinds of enzymes needed to be collected, based on the different bonds in the RNA of the viruses

Any virus is made up of RNA as their base structure

As the human DNA is made out of various organic compounds like C, H, O, N, all of them are in some order forms a certain strand, to form so they need to form a bond with each other, that makes up a molecule. There are various enzymes to break different bonds

- **Data Classification**

The collected data needs to be classified.

We need to collect a list of thousands of each kind of enzyme.

Each category needs to be put at a place with its respective labels.

After classifying and putting them in folders, we need to prepare a dataset on that.

Each kind of enzyme takes each label (label can be either numerical or a text).

Architecture

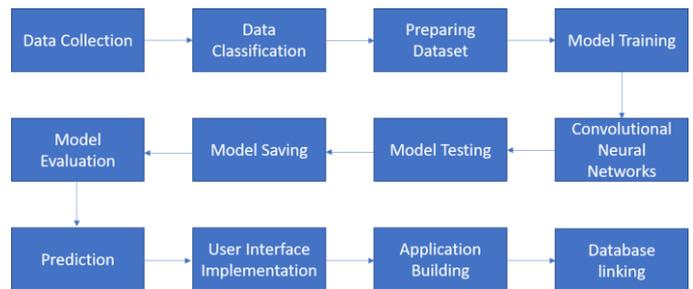


Figure-6: Architecture

The above architecture is self-explanatory. It indicates how the procedure is executed chronologically.

First, Data Collection is done. Once the data is collected properly, they are classified and categorized into separate labels.

Thus, classified data is called a Dataset. This image dataset is fed to the model that we will create using python programming using Deep Learning algorithms by implementing Convolutional Neural Networks.

The fed dataset is used as a training sample by the model. The training process takes a good amount of time, based on the number of pictures. We can use GPU for this process, this can happen using CUDA with Tensorflow. When the training is finished we proceed to the next step.

Convolutional Neural Networks are used for processing the data, the data undergoes through a lot of neural meshes and predicts a proper outcome, same as the number of data classes.

We go for the testing process. This involves the usage of a known image of the virus, chemical bond, and predicting the enzyme. The predictions are correct when the score is above 92% accuracy. In ML notations it is said that a model is efficient when the accuracy is more than 92%.

The model is saved using one of the two methods, either Joblib or Pickle, which are python libraries available for saving Machine Learning models. Joblib is used for saving complex models containing large data or complex data, like multi-dimensional arrays or images.

The next step is to evaluate the performance of the model. This evaluation is done explicitly by Tensorflow or Keras libraries of python. Based on the results we proceed to the next steps or correct the number of deep neural layers.

The model is saved using one of the two methods, either Joblib or Pickle, which are python libraries available for saving Machine Learning models. Joblib is used for saving complex models containing large data or complex data, like multi-dimensional arrays or images.

After the model is saved, we go for the testing process. This involves the usage of a known image of the virus, chemical bond, and predicting the enzyme. The predictions are correct when the score is above 92% accuracy. In ML notations it is said that a model is efficient when the accuracy is more than 92%.

The effective model is then exported. A simple user interface can be developed for engineers to use. This user interface can be accessed through an application. This application can be either Windows or Mac, based. This can also be made platform-independent.

Block Diagram

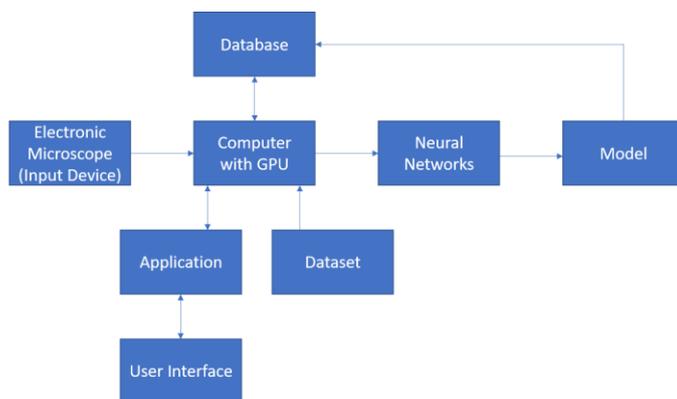


Figure-7: Block Diagram

REFERENCES

1. Bengio, Y.; Courville, A.; Vincent, P. (2013). "Representation Learning: A Review and New Perspectives". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 35 (8): 1798–1828. arXiv:1206.5538. doi:10.1109/tpami.2013.50. PMID 23787338. S2CID 393948.
2. ^ Jump up to:^{a b c d e f g h} Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". *Neural Networks*. 61: 85–117. arXiv:1404.7828. doi:10.1016/j.neunet.2014.09.003. PMID 25462637. S2CID 11715509.
3. ^ Bengio, Yoshua; LeCun, Yann; Hinton, Geoffrey (2015). "Deep Learning". *Nature*. 521 (7553): 436–444. Bibcode:2015Natur.521..436L. doi:10.1038/nature14539. PMID 26017442. S2CID 3074096.
4. ^ Hu, J.; Niu, H.; Carrasco, J.; Lennox, B.; Arvin, F. (2020). "Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning". *IEEE Transactions on Vehicular Technology*. 69 (12): 14413–14423. doi:10.1109/TVT.2020.3034800. S2CID 228989788. Archived from the original on 2020-11-16. Retrieved 2021-05-04.
5. ^ Jump up to:^{a b} Ciresan, D.; Meier, U.; Schmidhuber, J. (2012). "Multi-column deep neural networks for image classification". *2012 IEEE Conference on Computer Vision and Pattern Recognition*. pp. 3642–3649. arXiv:1202.2745. doi:10.1109/cvpr.2012.6248110. ISBN 978-1-4673-1228-8. S2CID 2161592.
6. ^ Jump up to:^{a b} Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffry (2012). "ImageNet Classification with Deep Convolutional Neural Networks" (PDF). *NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada*. Archived (PDF) from the original on 2017-01-10. Retrieved 2017-05-24.
7. ^ "Google's AlphaGo AI wins three-match series against the world's best Go player". *TechCrunch*. 25 May 2017. Archived from the original on 17 June 2018. Retrieved 17 June 2018.
8. ^ Marblestone, Adam H.; Wayne, Greg; Kording, Konrad P. (2016). "Toward an Integration of Deep Learning and Neuroscience". *Frontiers in Computational Neuroscience*. 10: 94. arXiv:1606.03813. Bibcode:2016arXiv160603813M. doi:10.3389/fncom.2016.00094. PMC 5021692. PMID 27683554. S2CID 1994856.
9. ^ Olshausen, B. A. (1996). "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". *Nature*. 381 (6583): 607–609. Bibcode:1996Natur.381..607O. doi:10.1038/381607a0. PMID 8637596. S2CID 4358477.
10. ^ Bengio, Yoshua; Lee, Dong-Hyun; Bornschein, Jorg; Mesnard, Thomas; Lin, Zhouhan (13 February 2015). "Towards Biologically Plausible Deep Learning". arXiv:1502.04156 [cs.LG].
11. ^ Schulz, Hannes; Behnke, Sven (1 November 2012). "Deep Learning". *KI - Künstliche Intelligenz*. 26 (4): 357–363. doi:10.1007/s13218-012-0198-z. ISSN 1610-1987. S2CID 220523562.
12. ^ Jump up to:^{a b c d e f} Deng, L.; Yu, D. (2014). "Deep Learning: Methods and Applications" (PDF). *Foundations and Trends in Signal Processing*. 7 (3–4): 1–199. doi:10.1561/2000000039. Archived (PDF) from the original on 2016-03-14. Retrieved 2014-10-18.
13. ^ Jump up to:^{a b c d e} Bengio, Yoshua (2009). "Learning Deep Architectures for AI" (PDF). *Foundations and Trends in Machine Learning*. 2 (1): 1–127. CiteSeerX 10.1.1.701.9550. doi:10.1561/22000000006. Archived from the original (PDF) on 4 March 2016. Retrieved 3 September 2015.
14. ^ LeCun, Yann; Bengio, Yoshua; Hinton, Geoffrey (28 May 2015). "Deep learning". *Nature*. 521 (7553): 436–444. Bibcode:2015Natur.521..436L. doi:10.1038/nature14539. PMID 26017442. S2CID 3074096.
15. ^ Shigeki, Sugiyama (12 April 2019). *Human Behavior and Another Kind in Consciousness: Emerging Research and Opportunities: Emerging Research and Opportunities*. IGI Global. ISBN 978-1-5225-8218-2.
16. ^ Bengio, Yoshua; Lamblin, Pascal; Popovici, Dan; Larochelle, Hugo (2007). Greedy layer-wise training of deep networks (PDF). *Advances in neural information processing systems*. pp. 153–160. Archived (PDF) from the original on 2019-10-20. Retrieved 2019-10-06.
17. ^ Jump up to:^{a b} Schmidhuber, Jürgen (2015). "Deep Learning". *Scholarpedia*. 10 (11): 32832. Bibcode:2015SchpJ..1032832S. doi:10.4249/scholarpedia.32832. Archived from the original on 2016-04-19. Retrieved 2016-04-09.
18. ^ Jump up to:^{a b c} Hinton, G.E. (2009). "Deep belief networks". *Scholarpedia*. 4 (5): 5947. Bibcode:2009SchpJ...4.5947H. doi:10.4249/scholarpedia.5947.
19. ^ Jump up to:^{a b c} Cybenko (1989). "Approximations by superpositions of sigmoidal functions" (PDF). *Mathematics of Control, Signals, and Systems*. 2 (4): 303–314. doi:10.1007/bf02551274. S2CID 3958369. Archived from the original (PDF) on 10 October 2015.
20. ^ Jump up to:^{a b c} Hornik, Kurt (1991). "Approximation Capabilities of Multilayer Feedforward Networks". *Neural Networks*. 4 (2): 251–257. doi:10.1016/0893-6080(91)90009-t.
21. ^ Jump up to:^{a b} Haykin, Simon S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall. ISBN 978-0-13-273350-2.
22. ^ Jump up to:^{a b} Hassoun, Mohamad H. (1995). *Fundamentals of Artificial Neural Networks*. MIT Press. p. 48. ISBN 978-0-262-08239-6.
23. ^ Jump up to:^{a b} Lu, Z., Pu, H., Wang, F., Hu, Z., & Wang, L. (2017). The Expressive Power of Neural Networks: A View from the Width Archived 2019-02-13 at the Wayback Machine. *Neural Information Processing Systems*, 6231-6239.
24. ^ [19][20][21][22][23]
25. ^ Jump up to:^{a b c d} Murphy, Kevin P. (24 August 2012). *Machine Learning: A Probabilistic Perspective*. MIT Press. ISBN 978-0-262-01802-9.
26. ^ [12][13][14][15][16][17][18][25]
27. ^ [19][20][21][22]

28. ^ Sonoda, Sho; Murata, Noboru (2017). "Neural network with unbounded activation functions is universal approximator". *Applied and Computational Harmonic Analysis*. 43 (2): 233–268. arXiv:1505.03654. doi:10.1016/j.acha.2015.12.005. S2CID 12149203.
29. ^ [12][13][1][2][18][25]
30. ^ Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. (2012). "Improving neural networks by preventing co-adaptation of feature detectors". arXiv:1207.0580[math.LG].
31. Copyright © 2021 Elsevier B.V. or its licensors or contributors. ScienceDirect ® is a registered trademark of Elsevier B.V.
32. <https://www.tensorflow.org/tutorials/images/classification>
33. Deng, L.; Abdel-Hamid, O.; Yu, D. (2013). "A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion" (PDF). Google.com. ICASSP. Archived (PDF) from the original on 2021-05-09. Retrieved 2017-06-13.
34. ^ Jump up to:^{a b} Sainath, Tara N.; Mohamed, Abdel-Rahman; Kingsbury, Brian; Ramabhadran, Bhuvana (2013). "Deep convolutional neural networks for LVCSR". *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. pp. 8614–8618. doi:10.1109/icassp.2013.6639347. ISBN 978-1-4799-0356-6. S2CID 13816461.
35. ^ Yann LeCun (2016). Slides on Deep Learning Online Archived 2016-04-23 at the Wayback Machine
36. ^ Jump up to:^{a b c} NIPS Workshop: Deep Learning for Speech Recognition and Related Applications, Whistler, BC, Canada, Dec. 2009 (Organizers: Li Deng, Geoff Hinton, D. Yu).
37. ^ Jump up to:^{a b} Keynote talk: Recent Developments in Deep Neural Networks. ICASSP, 2013 (by Geoff Hinton).
38. ^ D. Yu, L. Deng, G. Li, and F. Seide (2011). "Discriminative pretraining of deep neural networks," U.S. Patent Filing.
39. ^ Jump up to:^{a b c} Deng, L.; Hinton, G.; Kingsbury, B. (2013). "New types of deep neural network learning for speech recognition and related applications: An overview (ICASSP)" (PDF). Archived (PDF) from the original on 2017-09-26. Retrieved 2017-06-12.
40. ^ Jump up to:^{a b c} Yu, D.; Deng, L. (2014). *Automatic Speech Recognition: A Deep Learning Approach* (Publisher: Springer). ISBN 978-1-4471-5779-3.
41. ^ "Deng receives prestigious IEEE Technical Achievement Award - Microsoft Research". Microsoft Research. 3 December 2015. Archived from the original on 16 March 2018. Retrieved 16 March 2018.
42. ^ Jump up to:^{a b} Li, Deng (September 2014). "Keynote talk: 'Achievements and Challenges of Deep Learning - From Speech Analysis and Recognition To Language and Multimodal Processing'". *Interspeech*. Archived from the original on 2017-09-26. Retrieved 2017-06-12.
43. ^ Yu, D.; Deng, L. (2010). "Roles of Pre-Training and Fine-Tuning in Context-Dependent DBN-HMMs for Real-World Speech Recognition". *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*. Archived from the original on 2017-10-12. Retrieved 2017-06-14.
44. ^ Seide, F.; Li, G.; Yu, D. (2011). "Conversational speech transcription using context-dependent deep neural networks". *Interspeech*. Archived from the original on 2017-10-12. Retrieved 2017-06-14.
45. ^ Deng, Li; Li, Jinyu; Huang, Jui-Ting; Yao, Kaisheng; Yu, Dong; Seide, Frank; Seltzer, Mike; Zweig, Geoff; He, Xiaodong (1 May 2013). "Recent Advances in Deep Learning for Speech Research at Microsoft". *Microsoft Research*. Archived from the original on 12 October 2017. Retrieved 14 June 2017.
46. ^ "Nvidia CEO bets big on deep learning and VR". *Venture Beat*. 5 April 2016. Archived from the original on 25 November 2020. Retrieved 21 April 2017.
47. ^ "From not working to neural networking". *The Economist*. Archived from the original on 2016-12-31. Retrieved 2017-08-26.
48. ^ Jump up to:^{a b} Oh, K.-S.; Jung, K. (2004). "GPU implementation of neural networks". *Pattern Recognition*. 37 (6): 1311–1314. Bibcode:2004PatRe...37.1311O. doi:10.1016/j.patcog.2004.01.013.
49. ^ "A Survey of Techniques for Optimizing Deep Learning on GPUs Archived 2021-05-09 at the Wayback Machine", S. Mittal and S. Vaishay, Journal of Systems Architecture, 2019
50. ^ Jump up to:^{a b} Chellapilla, Kumar; Puri, Sidd; Simard, Patrice (2006). *High performance convolutional neural networks for document processing*, archived from the original on 2020-05-18, retrieved 2021-02-14
51. ^ Cireşan, Dan Claudiu; Meier, Ueli; Gambardella, Luca Maria; Schmidhuber, Jürgen (21 September 2010). "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition". *Neural Computation*. 22 (12): 3207–3220. arXiv:1003.0358. doi:10.1162/neco_a_00052. ISSN 0899-7667. PMID 20858131. S2CID 1918673.
52. ^ Raina, Rajat; Madhavan, Anand; Ng, Andrew Y. (2009). "Large-scale Deep Unsupervised Learning Using Graphics Processors". *Proceedings of the 26th Annual International Conference on Machine Learning. ICML '09*. New York, NY, USA: ACM: 873–880. CiteSeerX 10.1.1.154.372. doi:10.1145/1553374.1553486. ISBN 9781605585161. S2CID 392458.
53. ^ Sze, Vivienne; Chen, Yu-Hsin; Yang, Tien-Ju; Emer, Joel (2017). "Efficient Processing of Deep Neural Networks: A Tutorial and Survey". arXiv:1703.09039 [cs.CV].
54. ^ Jump up to:^{a b} "Merck Molecular Activity Challenge". kaggle.com. Archived from the original on 2020-07-16. Retrieved 2020-07-16.
55. ^ Jump up to:^{a b} "Multi-task Neural Networks for QSAR Predictions | Data Science Association". www.datascienceassn.org. Archived from the original on 30 April 2017. Retrieved 14 June 2017.
56. ^ Jump up to:^{a b} "Toxicology in the 21st century Data Challenge"

57. ^ Jump up to:^{a b} "NCATS Announces Tox21 Data Challenge Winners". Archived from the original on 2015-09-08. Retrieved 2015-03-05.
58. ^ Jump up to:^{a b} "Archived copy". Archived from the original on 28 February 2015. Retrieved 5 March 2015.
59. ^ Ciresan, D. C.; Meier, U.; Masci, J.; Gambardella, L. M.; Schmidhuber, J. (2011). "Flexible, High Performance Convolutional Neural Networks for Image Classification" (PDF). *International Joint Conference on Artificial Intelligence*. doi:10.5591/978-1-57735-516-8/ijcai11-210. Archived (PDF) from the original on 2014-09-29. Retrieved 2017-06-13.
60. ^ Ciresan, Dan; Giusti, Alessandro; Gambardella, Luca M.; Schmidhuber, Juergen (2012). Pereira, F.; Burges, C. J. C.; Bottou, L.; Weinberger, K. Q. (eds.). *Advances in Neural Information Processing Systems 25* (PDF). Curran Associates, Inc. pp. 2843–2851. Archived (PDF) from the original on 2017-08-09. Retrieved 2017-06-13.
61. ^ Ciresan, D.; Giusti, A.; Gambardella, L.M.; Schmidhuber, J. (2013). "Mitosis Detection in Breast Cancer Histology Images using Deep Neural Networks". *Proceedings MICCAI. Lecture Notes in Computer Science*. **7908** (Pt 2): 411–418. doi:10.1007/978-3-642-40763-5_51. ISBN 978-3-642-38708-1. PMID 24579167.
62. ^ Vinyals, Oriol; Toshev, Alexander; Bengio, Samy; Erhan, Dumitru (2014). "Show and Tell: A Neural Image Caption Generator". arXiv:1411.4555 [cs.CV].
63. ^ Fang, Hao; Gupta, Saurabh; Iandola, Forrest; Srivastava, Rupesh; Deng, Li; Dollár, Piotr; Gao, Jianfeng; He, Xiaodong; Mitchell, Margaret; Platt, John C; Lawrence Zitnick, C; Zweig, Geoffrey (2014). "From Captions to Visual Concepts and Back". arXiv:1411.4952 [cs.CV].
64. ^ Kiros, Ryan; Salakhutdinov, Ruslan; Zemel, Richard S (2014). "Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models". arXiv:1411.2539 [cs.LG].
65. ^ Zhong, Sheng-hua; Liu, Yan; Liu, Yang (2011). "Bilinear Deep Learning for Image Classification". *Proceedings of the 19th ACM International Conference on Multimedia. MM '11*. New York, NY, USA: ACM: 343–352. doi:10.1145/2072298.2072344. hdl:10397/23574. ISBN 9781450306164. S2CID 11922007.
66. ^ "Why Deep Learning Is Suddenly Changing Your Life". *Fortune*. 2016. Archived from the original on 14 April 2018. Retrieved 13 April 2018.
67. ^ Silver, David; Huang, Aja; Maddison, Chris J.; Guez, Arthur; Sifre, Laurent; Driessche, George van den; Schrittwieser, Julian; Antonoglou, Ioannis; Panneershelvam, Veda (January 2016). "Mastering the game of Go with deep neural networks and tree search". *Nature*. **529** (7587): 484–489. Bibcode:2016Natur.529..484S. doi:10.1038/nature16961. ISSN 1476-4687. PMID 26819042. S2CID 515925.
68. ^ *A Guide to Deep Learning and Neural Networks*, archived from the original on 2020-11-02, retrieved 2020-11-16
69. ^ Szegedy, Christian; Toshev, Alexander; Erhan, Dumitru (2013). "Deep neural networks for object detection". *Advances in Neural Information Processing Systems*: 2553–2561. Archived from the original on 2017-06-29. Retrieved 2017-06-13.
70. ^ Rolnick, David; Tegmark, Max (2018). "The power of deeper networks for expressing natural functions". *International Conference on Learning Representations. ICLR 2018*. Archived from the original on 2021-01-07. Retrieved 2021-01-05.
71. ^ Hof, Robert D. "Is Artificial Intelligence Finally Coming into Its Own?". *MIT Technology Review*. Archived from the original on 31 March 2019. Retrieved 10 July 2018.
72. ^ Jump up to:^{a b} Gers, Felix A.; Schmidhuber, Jürgen (2001). "LSTM Recurrent Networks Learn Simple Context Free and Context Sensitive Languages". *IEEE Transactions on Neural Networks*. **12** (6): 1333–1340. doi:10.1109/72.963769. PMID 18249962. Archived from the original on 2020-01-26. Retrieved 2020-02-25.
73. ^ Jump up to:^{a b c} Sutskever, L.; Vinyals, O.; Le, Q. (2014). "Sequence to Sequence Learning with Neural Networks" (PDF). *Proc. NIPS*. arXiv:1409.3215. Bibcode:2014arXiv1409.3215S. Archived (PDF) from the original on 2021-05-09. Retrieved 2017-06-13.
74. ^ Jump up to:^{a b} Jozefowicz, Rafal; Vinyals, Oriol; Schuster, Mike; Shazeer, Noam; Wu, Yonghui (2016). "Exploring the Limits of Language Modeling". arXiv:1602.02410 [cs.CL].
75. ^ Jump up to:^{a b} Gillick, Dan; Brunk, Cliff; Vinyals, Oriol; Subramanya, Amarnag (2015). "Multilingual Language Processing from Bytes". arXiv:1512.00103 [cs.CL].
76. ^ Mikolov, T.; et al. (2010). "Recurrent neural network based language model" (PDF). *Interspeech*. Archived (PDF) from the original on 2017-05-16. Retrieved 2017-06-13.
77. ^ Jump up to:^{a b} "Learning Precise Timing with LSTM Recurrent Networks (PDF Download Available)". *ResearchGate*. Archived from the original on 9 May 2021. Retrieved 13 June 2017.
78. ^ LeCun, Y.; et al. (1998). "Gradient-based learning applied to document recognition". *Proceedings of the IEEE*. **86** (11): 2278–2324. doi:10.1109/5.726791.
79. ^ Bengio, Yoshua; Boulanger-Lewandowski, Nicolas; Pascanu, Razvan (2013). "Advances in optimizing recurrent networks". *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. pp. 8624–8628. arXiv:1212.0901. CiteSeerX 10.1.1.752.9151. doi:10.1109/icassp.2013.6639349. ISBN 978-1-4799-0356-6. S2CID 12485056.
80. ^ Dahl, G.; et al. (2013). "Improving DNNs for LVCSR using rectified linear units and dropout" (PDF). *ICASSP*. Archived (PDF) from the original on 2017-08-12. Retrieved 2017-06-13.
81. ^ "Data Augmentation - deeplearning.ai | Coursera". *Coursera*. Archived from the original on 1 December 2017. Retrieved 30 November 2017.
82. ^ Hinton, G. E. (2010). "A Practical Guide to Training Restricted Boltzmann Machines". *Tech. Rep. UTML TR 2010-003*. Archived from the original on 2021-05-09. Retrieved 2017-06-13.

83. ^ You, Yang; Buluç, Aydın; Demmel, James (November 2017). "Scaling deep learning on GPU and knights landing clusters". *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis on - SC '17. SC '17*, ACM. pp. 1–12. doi:10.1145/3126908.3126912. ISBN 9781450351140. S2CID 8869270. Archived from the original on 29 July 2020. Retrieved 5 March 2018.
84. ^ Viebke, André; Memeti, Suejb; Pllana, Sabri; Abraham, Ajith (2019). "CHAOS: a parallelization scheme for training convolutional neural networks on Intel Xeon Phi". *The Journal of Supercomputing*. **75**: 197–227. arXiv:1702.07908. Bibcode:2017arXiv170207908v. doi:10.1007/s11227-017-1994-x. S2CID 14135321.
85. ^ Ting Qin, et al. "A learning algorithm of CMAC based on RLS." *Neural Processing Letters* 19.1 (2004): 49-61.
86. ^ Ting Qin, et al. "Continuous CMAC-QRLS and its systolic array Archived 2018-11-18 at the Wayback Machine." *Neural Processing Letters* 22.1 (2005): 1-16.
87. ^ Research, AI (23 October 2015). "Deep Neural Networks for Acoustic Modeling in Speech Recognition". *airesearch.com*. Archived from the original on 1 February 2016. Retrieved 23 October 2015.
88. ^ "GPUs Continue to Dominate the AI Accelerator Market for Now". *InformationWeek*. December 2019. Archived from the original on 10 June 2020. Retrieved 11 June 2020.
89. ^ Ray, Tiernan (2019). "AI is changing the entire nature of compute". *ZDNet*. Archived from the original on 25 May 2020. Retrieved 11 June 2020.
90. ^ "AI and Compute". *OpenAI*. 16 May 2018. Archived from the original on 17 June 2020. Retrieved 11 June 2020.
91. ^ TIMIT Acoustic-Phonetic Continuous Speech Corpus Linguistic Data Consortium, Philadelphia.
92. ^ Robinson, Tony (30 September 1991). "Several Improvements to a Recurrent Error Propagation Network Phone Recognition System". *Cambridge University Engineering Department Technical Report. CUED/F-INFENG/TR82*. doi:10.13140/RG.2.2.15418.90567.
93. ^ Abdel-Hamid, O.; et al. (2014). "Convolutional Neural Networks for Speech Recognition". *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. **22** (10): 1533–1545. doi:10.1109/taslp.2014.2339736. S2CID 206602362. Archived from the original on 2020-09-22. Retrieved 2018-04-20.
94. ^ Deng, L.; Platt, J. (2014). "Ensemble Deep Learning for Speech Recognition". *Proc. Interspeech*. S2CID 15641618.
95. ^ Tóth, László (2015). "Phone Recognition with Hierarchical Convolutional Deep Maxout Networks" (PDF). *EURASIP Journal on Audio, Speech, and Music Processing*. **2015**. doi:10.1186/s13636-015-0068-3. S2CID 217950236. Archived (PDF) from the original on 2020-09-24. Retrieved 2019-04-01.
96. ^ McMillan, Robert (17 December 2014). "How Skype Used AI to Build Its Amazing New Language Translator / WIRED". *Wired*. Archived from the original on 8 June 2017. Retrieved 14 June 2017.
97. ^ Hannun, Awni; Case, Carl; Casper, Jared; Catanzaro, Bryan; Diamos, Greg; Elsen, Erich; Prenger, Ryan; Satheesh, Sanjeev; Sengupta, Shubho; Coates, Adam; Ng, Andrew Y (2014). "Deep Speech: Scaling up end-to-end speech recognition". arXiv:1412.5567 [cs.CL].
98. ^ "MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges". *yann.lecun.com*. Archived from the original on 2014-01-13. Retrieved 2014-01-28.
99. ^ Cireşan, Dan; Meier, Ueli; Masci, Jonathan; Schmidhuber, Jürgen (August 2012). "Multi-column deep neural network for traffic sign classification". *Neural Networks. Selected Papers from IJCNN 2011*. **32**: 333–338. CiteSeerX 10.1.1.226.8219. doi:10.1016/j.neunet.2012.02.023. PMID 22386783.
100. ^ Nvidia Demos a Car Computer Trained with "Deep Learning" (6 January 2015), David Talbot, *MIT Technology Review*
101. ^ G. W. Smith; Frederic Fol Leymarie (10 April 2017). "The Machine as Artist: An Introduction". *Arts*. **6** (4): 5. doi:10.3390/arts6020005.
102. ^ Blaise Agüera y Arcas (29 September 2017). "Art in the Age of Machine Intelligence". *Arts*. **6** (4): 18. doi:10.3390/arts6040018.
103. ^ Goldberg, Yoav; Levy, Omar (2014). "word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method". arXiv:1402.3722 [cs.CL].
104. ^ Jump up to:^a ^b Socher, Richard; Manning, Christopher. "Deep Learning for NLP" (PDF). Archived (PDF) from the original on 6 July 2014. Retrieved 26 October 2014.
105. ^ Socher, Richard; Bauer, John; Manning, Christopher; Ng, Andrew (2013). "Parsing With Compositional Vector Grammars" (PDF). *Proceedings of the ACL 2013 Conference*. Archived (PDF) from the original on 2014-11-27. Retrieved 2014-09-03.
106. ^ Socher, Richard (2013). "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank" (PDF). Archived (PDF) from the original on 2016-12-28. Retrieved 2014-09-03.
107. ^ Shen, Yelong; He, Xiaodong; Gao, Jianfeng; Deng, Li; Mesnil, Gregoire (1 November 2014). "A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval". *Microsoft Research*. Archived from the original on 27 October 2017. Retrieved 14 June 2017.
108. ^ Huang, Po-Sen; He, Xiaodong; Gao, Jianfeng; Deng, Li; Acero, Alex; Heck, Larry (1 October 2013). "Learning Deep Structured Semantic Models for Web Search using Clickthrough Data". *Microsoft Research*. Archived from the original on 27 October 2017. Retrieved 14 June 2017.
109. ^ Mesnil, G.; Dauphin, Y.; Yao, K.; Bengio, Y.; Deng, L.; Hakkani-Tur, D.; He, X.; Heck, L.; Tur, G.; Yu, D.; Zweig, G. (2015). "Using recurrent neural networks for slot filling in spoken language understanding". *IEEE Transactions on Audio, Speech, and Language Processing*. **23** (3): 530–539. doi:10.1109/taslp.2014.2383614. S2CID 1317136.
110. ^ Jump up to:^a ^b Gao, Jianfeng; He, Xiaodong; Yih, Scott Wen-tau; Deng, Li (1 June 2014). "Learning Continuous Phrase Representations for Translation

- Modeling*". Microsoft Research. Archived from the original on 27 October 2017. Retrieved 14 June 2017.
111. ^ Brocardo, Marcelo Luiz; Traore, Issa; Woungang, Isaac; Obaidat, Mohammad S. (2017). "Authorship verification using deep belief network systems". *International Journal of Communication Systems*. **30** (12): e3259. doi:10.1002/dac.3259.
 112. ^ "Deep Learning for Natural Language Processing: Theory and Practice (CIKM2014 Tutorial) - Microsoft Research". Microsoft Research. Archived from the original on 13 March 2017. Retrieved 14 June 2017.
 113. ^ Turovsky, Barak (15 November 2016). "Found in translation: More accurate, fluent sentences in Google Translate". *The Keyword Google Blog*. Archived from the original on 7 April 2017. Retrieved 23 March 2017.
 114. ^ Jump up to:^{a b c d} Schuster, Mike; Johnson, Melvin; Thorat, Nikhil (22 November 2016). "Zero-Shot Translation with Google's Multilingual Neural Machine Translation System". *Google Research Blog*. Archived from the original on 10 July 2017. Retrieved 23 March 2017.
 115. ^ Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory". *Neural Computation*. **9** (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276. S2CID 1915014. Archived from the original on 2021-01-22. Retrieved 2017-10-12.
 116. ^ Felix A. Gers; Jürgen Schmidhuber; Fred Cummins (2000). "Learning to Forget: Continual Prediction with LSTM". *Neural Computation*. **12** (10): 2451–2471. CiteSeerX 10.1.1.55.5709. doi:10.1162/08997660030015015. PMID 11032042. S2CID 11598600.
 117. ^ Wu, Yonghui; Schuster, Mike; Chen, Zhifeng; Le, Quoc V; Norouzi, Mohammad; Macherey, Wolfgang; Krikun, Maxim; Cao, Yuan; Gao, Qin; Macherey, Klaus; Klingner, Jeff; Shah, Apurva; Johnson, Melvin; Liu, Xiaobing; Kaiser, Łukasz; Gouws, Stephan; Kato, Yoshikiyo; Kudo, Taku; Kazawa, Hideto; Stevens, Keith; Kurian, George; Patil, Nishant; Wang, Wei; Young, Cliff; Smith, Jason; Riesa, Jason; Rudnick, Alex; Vinyals, Oriol; Corrado, Greg; et al. (2016). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". *arXiv:1609.08144 [cs.CL]*.
 118. ^ Metz, Cade (27 September 2016). "An Infusion of AI Makes Google Translate More Powerful Than Ever". *Wired*. Archived from the original on 8 November 2020. Retrieved 12 October 2017.
 119. ^ Jump up to:^{a b} Boitet, Christian; Blanchon, Hervé; Seligman, Mark; Bellyncq, Valérie (2010). "MT on and for the Web" (PDF). Archived from the original (PDF) on 29 March 2017. Retrieved 1 December 2016.
 120. ^ Arrowsmith, J; Miller, P (2013). "Trial watch: Phase II and phase III attrition rates 2011-2012". *Nature Reviews Drug Discovery*. **12** (8): 569. doi:10.1038/nrd4090. PMID 23903212. S2CID 20246434.
 121. ^ Verbist, B; Klambauer, G; Vervoort, L; Talloen, W; The Qstar, Consortium; Shkedy, Z; Thas, O; Bender, A; Göhlmann, H. W.; Hochreiter, S (2015). "Using transcriptomics to guide lead optimization in drug discovery projects: Lessons learned from the QSTAR project". *Drug Discovery Today*. **20** (5): 505–513. doi:10.1016/j.drudis.2014.12.014. PMID 25582842.
 122. ^ Wallach, Izhar; Dzamba, Michael; Heifets, Abraham (9 October 2015). "AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery". *arXiv:1510.02855 [cs.LG]*.
 123. ^ "Toronto startup has a faster way to discover effective medicines". *The Globe and Mail*. Archived from the original on 20 October 2015. Retrieved 9 November 2015.
 124. ^ "Startup Harnesses Supercomputers to Seek Cures". *KQED Future of You*. Archived from the original on 24 December 2015. Retrieved 9 November 2015.
 125. ^ "Toronto startup has a faster way to discover effective medicines". Archived from the original on 2015-12-25. Retrieved 2017-08-26.
 126. ^ Gilmer, Justin; Schoenholz, Samuel S.; Riley, Patrick F.; Vinyals, Oriol; Dahl, George E. (2017-06-12). "Neural Message Passing for Quantum Chemistry". *arXiv:1704.01212 [cs.LG]*.
 127. ^ Zhavoronkov, Alex (2019). "Deep learning enables rapid identification of potent DDR1 kinase inhibitors". *Nature Biotechnology*. **37** (9): 1038–1040. doi:10.1038/s41587-019-0224-x. PMID 31477924. S2CID 201716327.
 128. ^ Gregory, Barber. "A Molecule Designed By AI Exhibits 'Druglike' Qualities". *Wired*. Archived from the original on 2020-04-30. Retrieved 2019-09-05.
 129. ^ Tkachenko, Yegor (8 April 2015). "Autonomous CRM Control via CLV Approximation with Deep Reinforcement Learning in Discrete and Continuous Action Space". *arXiv:1504.01840 [cs.LG]*.