MANTECH Publications

Journal of Artificial Intelligence, Machine Learning and Soft Computing
Volume 3, Issue 1, January-June 2018

Review: Convolutional Neural Network Machine Learning Algorithm

Nagesh Kumar D N[1], Madhukar M[2], Kavyashree M[3], Abhisht R Shetty[4]

Department of Electronics and Communication Engineering

Jyothy Institute of Technology, Bangalore, India

***Email ID**: nageshjit@gmail.com, madhu8703@gmail.com,] kavyashree.m195@gmail.com, abhisht.shetty@gmail.com*

Abstract

In machine learning, a convolutional neural network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery.

This paper, proposes the possibility of learning deep network structures and the relative study of their wide range of applications. CNNs use a variation of multilayer perceptron designed to require minimal preprocessing. The engineering of CNN is accomplished with neighborhood associations and tied weights overtaken by some type of pooling which brings about interpretation invariant highlights. Another advantage of CNNs is that they are less demanding to prepare and have numerous less parameters than completely associated systems with a similar number of concealed units. An Artificial Neural Network (ANN) with numerous concealed layers between the info and yield layers is a profound neural system (DNN). Deep learning structures of CNN, have been connected to fields of computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation and bioinformatics where the delivered comes about tantamount to and at times better than human specialists.

Keywords: *Neural Network (CNN), Field-Programmable Gate Array (FPGA), Deep Neural Networks, Drop-out algorithm.*

INTRODUCTION

Convolutional systems were enlivened by natural procedures in which the accessibility architecture between neurons is propelled by the relationship of the creature visual cortex. Individual cortical neurons react to boosts just in a confined locale of the visual field known as the open field. The responsive fields of various neurons incompletely cover with the end goal that they cover the whole visual field.

CNNs utilize generally little pre-handling contrasted with other picture order calculations. This implies the system takes in the channels that in customary calculations were hand-built. This freedom from earlier learning and human exertion in highlight configuration is a noteworthy preferred standpoint. Each convolutional neuron forms information just for its responsive field. Tiling enables CNNs to endure interpretation of the information picture (eg. translation, rotation, perspective distortion). Albeit completely associated feedforward neural systems can be utilized to learn includes and also characterize information, it isn't down to earth to apply this design to pictures. A high number of neurons would be essential even in a shallow design (opposite of deep). The convolution operation conveys an answer for this issue as it diminishes the quantity of free parameters, enabling the system to be more profound with less parameters. at the end of the day, it settle the vanishing or detonating inclinations issue in preparing conventional multi-layer neural systems with many layers by utilizing backpropagation. It is believed that deeper networks produce better recognition results. There is a lot of research on how to improve the performance of genetic algorithms for various applications.

The remainder of this paper is organized applications of CNN as follows:

Section 2 illustrates, An FPGA-Based Stream Processor for real time vision and the concept of Angel-Eye. Section 3 describes the Memory-Centric Accelerator Design and Embedded Streaming Deep Neural Networks Accelerator. Section 4 details out Hardware Design Automation, CNN for Human Activity Recognition using Mobile Sensors. Section 5 presents, Facial image processing with Convolutional Neural Networks and Medical Image Classification with Convolutional Neural Network.

FPGA BASED APPLICATIONS OF CNN

A. A Stream Processor with FPGA for embedded real time vision.

Numerous current visual recognition frameworks can be viewed as being made out of different layers of convolutional channel banks, scattered with different sorts of non-linearities. This incorporates Convolutional Networks, HMAX-sort designs, and in addition frameworks in light of thick SIFT highlights or Histogram of Gradients. Vision frameworks have advanced a ton in the previous decade, yet the vast majority of the cutting edge calculations still require a measure of calculation that makes their coordination to independent vehicles, cameras or toys unimaginable. This work is a stage towards low power, lightweight, and ease vision frameworks that are required for such applications[1].

This portrays an execution of an entire vision framework on a solitary Field-Programmable Gate Array (FPGA). The plan requires no outside equipment, other than a memory chip, and has been coordinated on a little 70 × 80 mm PCB, which devours under 15W, with a camera. The framework is programmable, and is able to execute all types of vision framework in which the main part of the calculation is gone through on convolutions with little size bits. The plan is particularly equipped towards. Convolutional Networks, yet can be utilized for some comparable structures in light of nearby channel banks and classifiers, for example, HMAX and HoG strategies.

B: Mapping CNN onto Embedded FPGA.

The calculation multifaceted nature of CNN is considerably more than conventional calculations. With assistance of GPU speeding up, applications of CNN are broadly sent in servers. In any case, for installed stages, CNN-based arrangements are still more complex to be connected.

In this paper, the examination of best in class models of CNN and its applications has been finished. Angel-Eye which is a programmable and adaptable CNN accelerator engineering, together with information quantization procedure and accumulation apparatus is discussed[2]. Information quantization technique lessens the bit-width down to 8-bit with irrelevant exactness misfortune. The assemblage instrument maps a specific CNN model proficiently onto hardware.

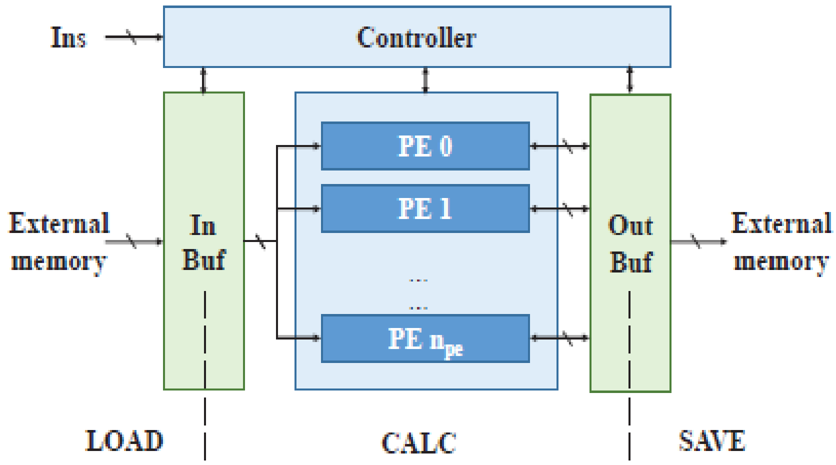

Figure 1: Overall architecture of Angel-Eye

Figure 1 shows the hardware design to help this guideline interface. It can be separated into four sections: PE array, On-chip Buffer, External Memory and Controller.

PE Array: The PE array actualizes the convolution regulations in CNN. Three levels of correspondence are executed by PE array:

1. **Kernel stage correspondence**: Every PE comprises of a few convolution motors. Every convolution motor processes the internal result of the convolution portion and a window of the picture in parallel.

2. **Input channel correspondence**: Distinctive convolution motors in every PE do convolution on various information directs in parallel. The aftereffects of various info channels are included as CNN characterizes.

3. **Output channel correspondence:** Diverse PEs share a similar information channels, yet not the convolution bits, to figure distinctive yield diverts in parallel. In this paper, An information quantization methodology is introduced to pack the bit-width utilized as a part of CNN. Assessed on state-of-the-art CNN design, this procedure carries insignificant execution with 16-bit and 8-bit setup. A compiler is likewise actualized to outline CNN models to guideline arrangements. Enhancement is done on arrangement to completely use the on-chip store and the parallelism amongst computation and information I/O.Units

ACCELERATOR DESIGNS FOR CNN

Memory-Centric Accelerator Design for Convolutional Neural Networks.

For portability and protection reasons, the required picture preparing ought to be nearby on implanted PC stages with execution prerequisites and vitality imperatives. Committed increasing speed of Convolutional Neural Networks (CNN) can accomplish these objectives with enough adaptability to play out different vision undertakings. A testing issue for the plan of effective quickening agents is the constrained measure of outer memory transmission capacity. The installed preparing by implanted PC stages needs to fulfill continuous execution necessities while obliging vitality use, and ought to be sufficiently adaptable to help numerous applications, which is at present not yet the situation.

This paper exhibits a memory-driven quickening agent to enhance execution without expanding memory data transfer capacity[3]. This quickening agent utilizes particular recollections that help the information development designs and streamlined booking for information region. This mix enables the required cushion size to be limited and information reuse to be amplified. In particular, we make the accompanying commitments:

- A configurable quickening agent layout for CNN, with adaptable information reuse supports. The layout bolsters the distinctive register designs in the CNN workload and can be arranged to coordinate the outside memory data transmission.

- A memory-driven plan stream to combine and program the quickening agent format. Our outline stream utilizes snappy plan space investigation to enhance on-chip memory size and information reuse.

- A high-level confirmation and assessment of the strategy by FPGA mapping of a speed movement sign acknowledgment application.

This empowers the blend of quickening agents that are exceptionally effective as far as use, FPGA assets and outside data transfer capacity prerequisites. Contrasted with quickening agents with standard scratchpad recollections, the cradle assets can be lessened up to a factor 13 while looking after execution. Also, when a similar measure of FPGA assets is utilized our quickening agents are up to 11 times quicker.

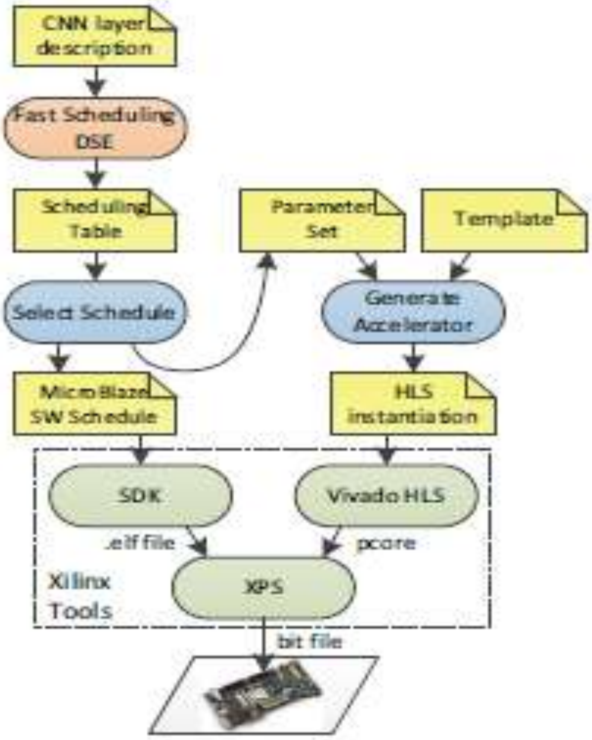

Figure 2: Accelerator design, mapping and configuration flow.

To instantiate the accelerator template, a HW/SW incorporation stream is utilized. Figure 2 gives an outline of the stream that arranges the HW format. The upper left part contains the planning configuration space investigation, from which the ideal calendars are utilized to choose the format parameters. The parameter set and the equipment format are physically changed over into a HLS instantiation of the quickening agent.

In the left piece of the plan stream, the chose plan is physically changed over to control programming for the MicroBlaze have processor. The accelerator template and the design flow are combined into an end-to-end arrangement that radically decreases improvement time for proficient CNN accelerators for an inserted vision stage. This outcome is examined and confirmed broadly on a Virtex 6 FPGA board with the Xilinx device chain. For the investigation, a CNN vision application for speed sign acknowledgment on a 720p HD video is utilized.

The CNN model does not change over fluctuating vision applications, but only the system setup and weight coefficients change if an alternate application is begun.

Embedded Streaming Deep Neural Networks Accelerator with Applications

ARTIFICIAL vision systems mean to give visual comprehension from crude pictures, i.e., change over high-dimensional information, for example, pictures and recordings, into valuable low-dimensional information, where choices can be made[4]. DCNNs are effective instruments that utilization many layers and channels to separate significant highlights of articles in pictures in a progressive way. They comprise of different layers of convolutions, each including in the vicinity of several channels. This paper exhibits a profoundly improved control technique for the versatile equipment engineering for DCNNs. The control technique goes about as a compiler and changes high-level representations of DCNNs into operation codes with the end goal of executing applications inside an equipment quickening agent. The control strategy exploits the attributes of DCNNs and investigates the computational energy of the custom equipment.

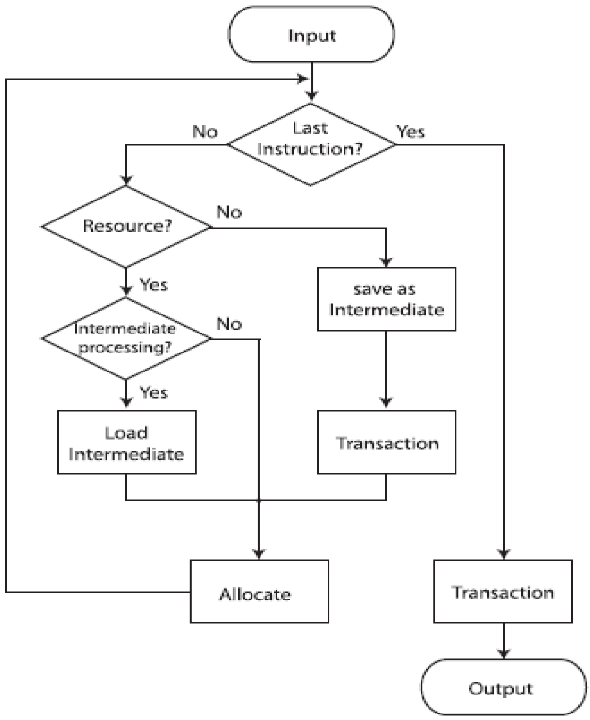

Figure. 3. Flowchart of the control method.

The flowchart of the system is shown in Fig. 3. The table containing columns of guidelines from the system parser is planned line by push (as long as assets allow). At the point when assets are distributed, every 50% of the accumulations is allotted for each arrangement of channels. For whatever length of time that ports and accumulations are accessible, assets keep on being apportioned, and once one sort of asset is totally allotted, the exchange is performed. On the off chance that the estimations have not been finished for a yield highlight outline, yields of the exchange are spared as moderate esteems. In the following cycle, the framework checks if there are middle of the road esteems that should be prepared further. Assuming this is the case, it allots assets for them too. This directing plan is extremely adaptable and can process DCNNs with any number of filters and layers.

Real-Time Automation applications of CNN

Hardware Design Automation of Convolutional Neural Networks

CNNs are at present the best in class approach for picture acknowledgment and grouping, having a place with the class of Supervised Learning. The thought behind the CNN calculation is to duplicate the system in the essential visual cortex of living beings; specifically, these phones are orchestrated in open fields that catch data from nearby districts of the field of view. This work proposes a structure implied as a device for the client to quicken and disentangle the plan and the usage of CNNs on FPGAs by utilizing High Level Synthesis, as yet giving a specific level of customization of the equipment outline[5]. The work exhibited here permits to accomplish a decent expectation exactness for the utilization of various assets accessible on the FPGA, taking into consideration a forecast blunder of under 30 BRAMs, 800 LUTs, and 1400 FFs, for a large portion of the experiments.

This work displays an electronic system that permits to outline and redo a CNN by methods for an online

Graphical User Interface (GUI). The customer side has been executed in HTML5 and Javascript, while the back-end is composed in Python. The work process of the application is appeared in Figure 5. The system requires an abnormal state detail of the system and a record containing the prepared weights as inputs.

In the accompanying subsections the different times of the work procedure are shown, while a full depiction of the framework can be found in the customer must decide the key structure of the CNN regardless, by giving the amount of convolutional and straight layers, and the estimation of the data that the framework will process. Another information required to create the CNN code is the game plan of weights of the particular layers. The client can choose either to transfer his own particular record, which can be traded with little exertion from Machine Learning systems accessible on the web. It is additionally conceivable to let the application to produce arbitrary esteems for the weights, with the goal that he can assess the equipment usage of the system concentrating on execution and assets.

At that point, by mean of a graphical representation of the system, each layer can be tweaked by determining its parameters. For convolutional layers, the client can set the quantity of channels and their measurement. Additionally, a sub-testing layer actualizing Max-or Mean-pooling can be incorporated and arranged at this stage. For direct layers, the client can determine the quantity of neurons in the system.

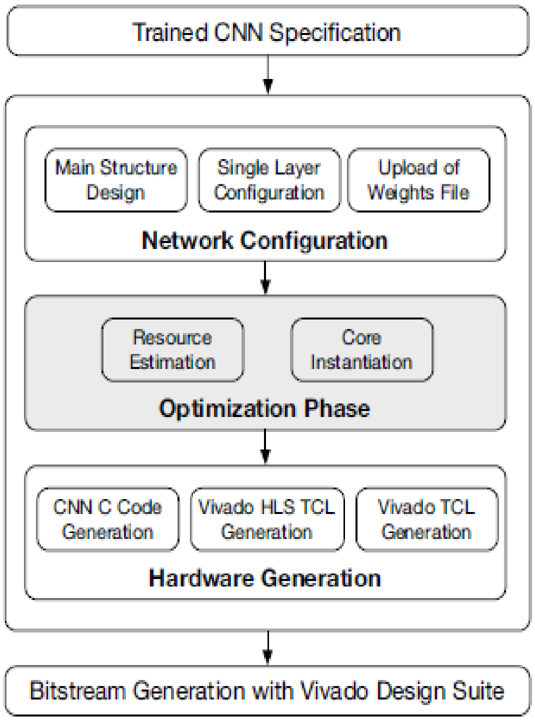

Figure 5: Workflow of the framework

Now, the back-end of the application will create the C++ source code executing the CNN and the tcl contents required by the Xilinx Tools as yield. The tcl contents are utilized to incorporate the entire plan up to the age of the bitstream with Vivado and Vivado HLS. As respects to the HLS of the CNN center, we misused the dataflow example of the calculation and we added mandates to the C++ source code produced by the structure. Since systems of various measurements and arrangements can be enhanced with case-particular approach.

Table 1: Summary of the resource estimation results.

Resource	Avg. Abs. Error	Std. Abs Error	Avg. Rel. Error [%]	Std. Rel. Error [%]
BRAM	15.82	13.38	24.41	22.53
LUT	430.28	236.09	3.71	1.83
FF	1263.88	182.63	14.88	1.47
DSP	2	0	1.96	0.07

Asset estimation has been done by applying the models portrayed in the past segment to each of the layers making the systems, and after that summing up the expectation for each kind of asset. Table II reports the outcomes acquired. As should be obvious from the tables, LUTs and DSPs are anticipated extremely well. LUTs have a normal relative forecast mistake of 3.71% with a low change, and DSPs aren't right anticipated by 2 units in every one of the tests. FFs are anticipated with a normal relative blunder of 15% with a low difference. In any case, the supreme mistake is under 30 BRAMs in over 80% of the tested CNN.

Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors.

An assortment of genuine portable detecting applications are getting to be plainly accessible, particularly in the life-logging, wellness following and wellbeing checking areas. These applications utilize portable sensors implanted in advanced mobile phones to perceive human exercises keeping in mind the end goal to improve comprehension of human conduct. While advance has been made, human action acknowledgment remains a testing errand. This paper proposes a way to deal with automatically extract discriminative highlights for movement acknowledgment[6].

In particular, we build up a technique in view of Convolutional Neural Networks (CNN), which can catch neighborhood reliance and scale invariance of a flag as it has been appeared in discourse acknowledgment and picture acknowledgment areas. Furthermore, an adjusted weight sharing strategy, called

MANTECH Publications
Journal of Artificial Intelligence, Machine Learning and Soft Computing
Volume 3, Issue 1, January-June 2018

halfway weight sharing, is proposed and connected to accelerometer signs to get further enhancements.

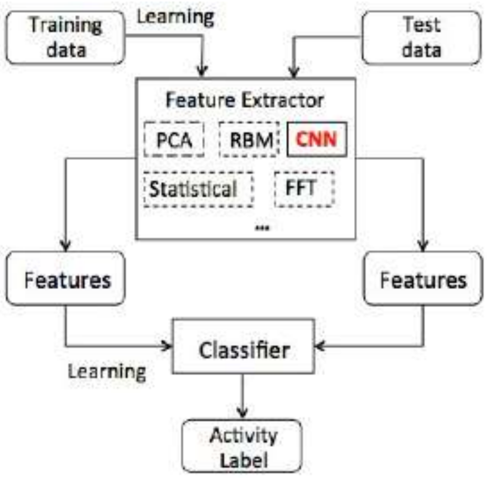

Figure 6: Feature extraction is one of the key components of activity recognition.

Figure 6 shows the activity recognition process, which is divided into training phase and classification phase. In the training phase, we extract features from the raw time series data. These features are then used to train a classification model. In the classification phase, we first extract features from unseen raw data and then use the trained prediction model to predict an activity label. Statistical features such as mean, standard deviation, entropy and correlation coefficients, etc. are the most widely used hand-crafted features in AR.

Fourier transform and wavelet transform are another two commonly used hand-crafted features, while discrete cosine transform (DCT) have also been applied with promising results, as well as auto-regressive model coefficients. Recently, time-delay embeddings have been applied for activity recognition. It adopts nonlinear time series analysis to extract features from time series and shows a significant improvement on periodic activities recognition (cycling involves a periodic, roughly two-dimensional leg movement). However, the features from time-delay embedding are less suitable for non-periodic activities.

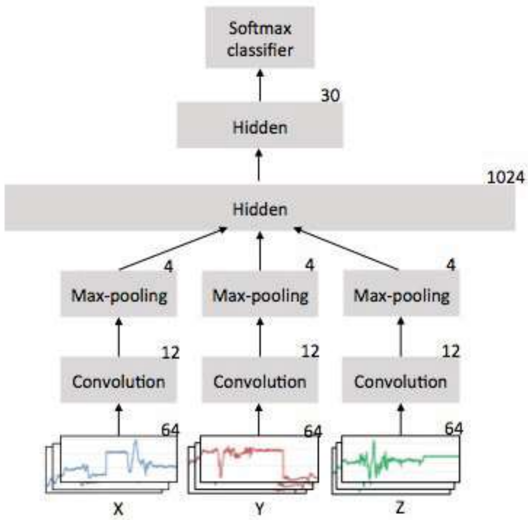

Figure 7: Structure of CNN for Human Activity Recognition. The dimension of input data is 64, the dimension convolutional output is 12 and the dimension max-pooling output is 4. The dimension of two hidden layers is 1024 and 30, respectively. The top layer is a Softmax classifier.

Figure 7 shows the structure of the proposed approach.

In order to capture the local dependencies of the data, one can enforce a local connectivity constraint between units of adjacent layers. For example, in Fig 8 the units (neurons) in the middle layer are only connected to a local subset of units in the input layer. From biology, we know that there are complex arrangement of cells in visual cortex, which are sensitive to small regions of the input, called a receptive field, and are tiled to generate the entire visual field. These filters are local in input space and are thus suited to exploit local correlation hidden in the data, so we also call it local filter. In terms of local filter, the weight of edge connected ith unit with jth, wi,j can be reduced by wa, and wi,j = wi,j+m = wa, where m is the width of the local filter.

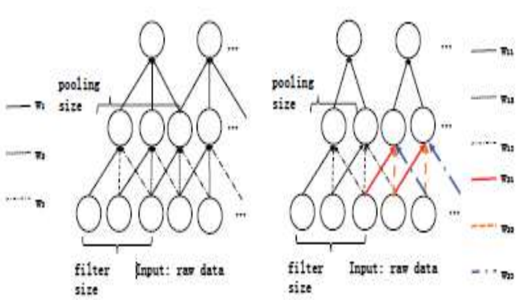

Figure 8: (Left) Traditional weight sharing CNN, (Right) Partial weight sharing CNN. Weights denoted by the same line style are shared

The topological constraint corresponds to learning a weight matrix with sparsity constraint, which is not only good for extracting local dependencies, but also reduces the computational complexity. The output of such a set of local filters constitute a feature map. At each temporal position, different types of units in different feature maps compute different types of features. The experimental results have shown that by extracting these characteristics, the CNN-based approach outperforms the state of-the-art approaches.

CNN FOR IMAGE PROCESSING

Facial image processing with Convolutional Neural Networks

Facial image preparing is a scope of research focused on the extraction and examination of information about human faces, information which is seemed to assume a central part in social communications including acknowledgment, feeling and goal. Facial investigation has different applications in inserted systems[7].

Among them we can refer to programmed center in computerized cameras, upgraded versatile videoconference, biometry, and canny UIs, show based video coding, picture recovery, reconnaissance and biometrics, and shrewd human-PC collaboration.

Figure 9 presents the general engineering of a ConvNets, which comprises in a pipeline of convolution, subsampling and neuronal operations. This pipeline performs programmed include extraction in an info retina, and the grouping of the extricated highlights, in a solitary coordinated plan. The three principle sorts of layers in ConvNets are:

1. Layers Ci are known as convolutional layers, and have a specific number of planes. Every component in a plane gets a contribution from a little neighborhood in the planes of the past layer. Each plane can be considered as a component outline with a settled element indicator. An inclination is added to the aftereffects of each convolutional veil. Different planes are utilized as a part of each layer with the goal that numerous highlights can be distinguished.

2. Layers Si are known as subsampling layers and comprise in neighborhood averaging and subsampling methods. This subsampling method isolates by two each measurement of the info outline expands the degrees of invariance to interpretation, scale, and twisting of the learnt designs.

3. Layers Ni are the order layers, and are connected after element extraction and information dimensionality lessening of Ci and Si layers. These layers compare to a multilayer perceptron. Convolutional Face Finder (CFF) which can limit various appearances with a base size of 20 pixels, pivoted up to plus or minus 20^0 in the picture plane and swung up to plus or minus 60^0.

The CFF (Fig. 9-a) consists in “four 5×5 convolution maps in C1 layer, fourteen 3×3 maps in C2 layer, 14 neurons in N1 layer, and a single neuron in N2 layer classifying input retina as face or non-face.

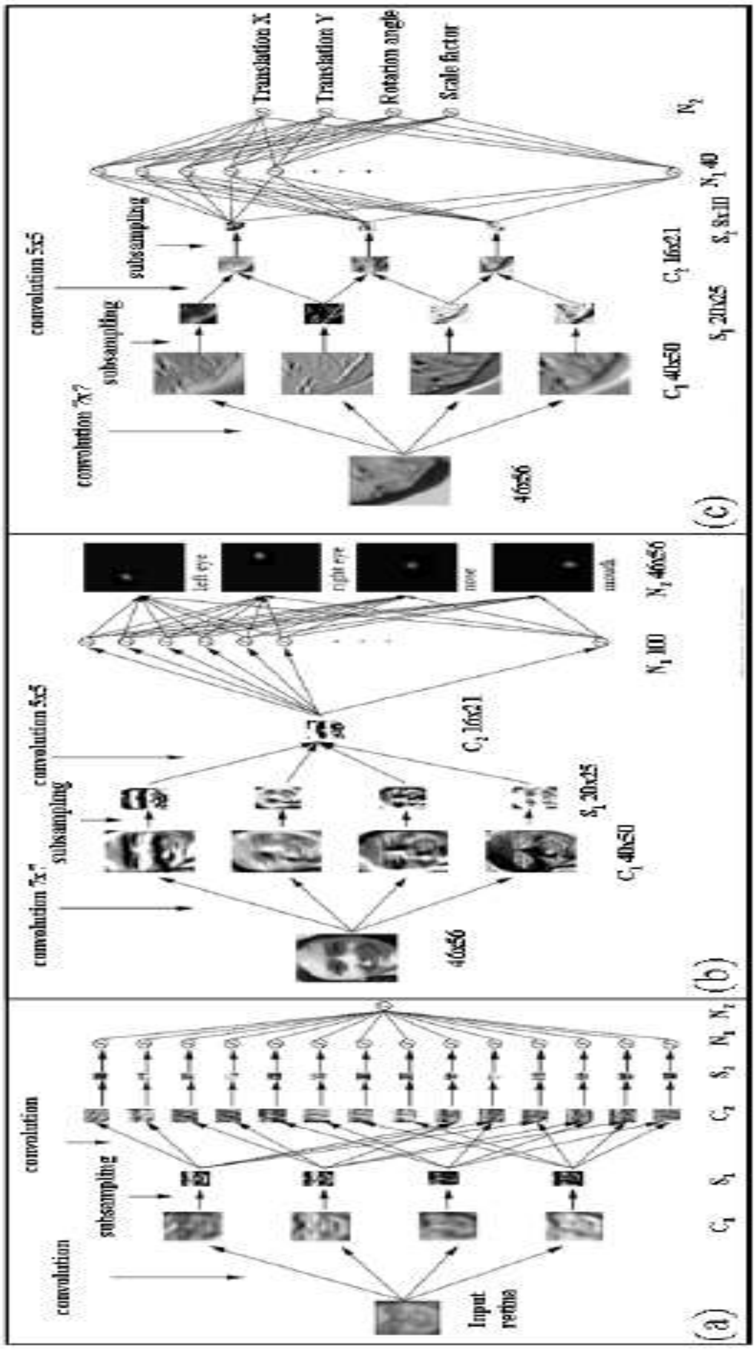

Figure 9: The facial interpretation Convolutional Neural Network framework: (a) CFF face detector, (b) Facial characteristic analyzer, (c) Face aligner

The architecture of this ConvNet is presented in Fig. 9-b: it has no S2 layer, and 4 output maps consisting of arrays of neurons with the same dimensions as the retina. C1 layer comprises four 7×7 convolution maps, C2 a single 5×5 convolution map, N1 a hundred neurons connected to the C2 feature map. A global face alignment method (Fig. 9-c) consists of a network very similar to the face detector, using four 7×7 convolution maps in C1, three 5×5 maps in C2, and four output neurons. These neurons are trained to estimate simultaneously translation in both axis, rotation and scale parameters of the transformation the input face image has undergone. A nonaligned detected face is then presented to the network which produces an estimation of the underlying transformation. To improve the correction an iterative scheme is applied where at each iteration only a certain portion (10%) of the correction is applied to the bounding box giving a new input retina and 94% of the face images of the BioID database" are obtained.

The CFF algorithm has been implemented on this platform and optimized accordingly, and Table II shows a correlation between the first and improved adaptation for QCIF image processing. These optimizations enables to reach a speed up factor up to 700, and to propose an embedded library which can robustly detect, track, align and recognize facial features on mobile phones at up to 3 QCIF frames per second. **See Table: 2**

Convolutional Neural Network in Medical Image analysis

In image analysis issues, the illustration and discriminative power highlights extricated are basic to get great characterization execution. This work proposes a modified CNN arrange for programmed lung picture fix grouping on neural-based machine learning structure to extricate discriminative highlights from preparing tests and perform characterization at the same time[8]. This has likewise joined arbitrary neural hub drop-out and utilized a solitary convolutional layer design to decrease the quantity of parameters in the CNN model to stay away from the over-fitting issue.

Table 2: Reference and Optimized CFF Performances Comparison

	Reference	**Optimized**
Detection Rate on CMU database	87.99%	88.2%
Processing rate	0.3 fr./s	6.5 fr./s
Memory footprint	3800 kByte	220 kByte

MANTECH Publications
Journal of Artificial Intelligence, Machine Learning and Soft Computing
Volume 3, Issue 1, January-June 2018

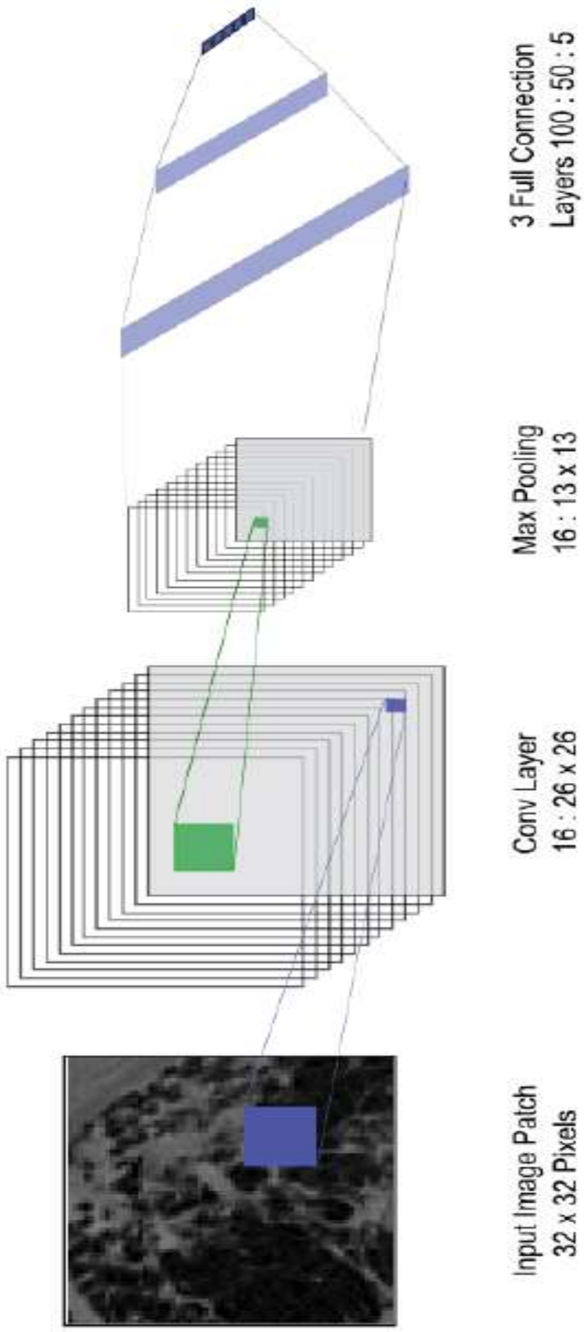

Figure 10: Network architecture of CNN for image classification

The overall framework design is illustrated in Figure 10. Contribution of the system is standardized lung picture fix with unit difference and zero mean. The main layer is a convolutional layer with part size of 7 × 7 pixels and 16 yield channels. The second layer is a maximum pooling layer with 2×2 bit measure. The accompanying three layers are completely associated neural layers with 100-50-5 neurons in each layer. The system with a solitary convolutional layer executes in the same class as systems with different convolution layers in ILD lung picture fix characterization errands. Picture pixels could be specifically utilized as contribution to standard encourage forward neural systems to determine picture arrangement issues. CNN models join weights into substantially lesser piece channels that rearrange the learning model. Drop-out calculation is connected to enhance the execution, by haphazardly crippling neurons in each layer amid preparing. “A drop-out map with the same size of the neurons in each layer is randomly initialized to mark the on or off state of the corresponding neuron at the start of each training iteration. The neurons with off state are then removed from the network during the training iteration, by disabling the activation signal forward propagation and error signal backward propagation of the neuron. It is equivalent to switching between different models for each learning iteration, so that many different models are trained at the same time”. During testing, all neurons are turned on, but with the activation signal attenuated to the probability of average turn on rate during the training phase.

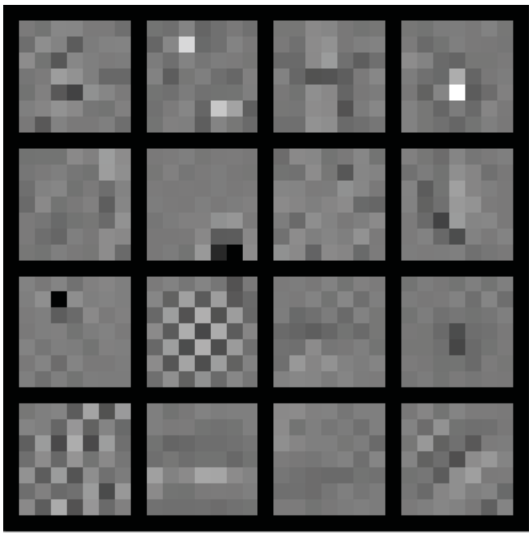

***Figure 11: DCT-like kernel matrices visualized as small picture patches*.**

The kernel matrices in the convolutional Layer represent sets of features learned by the network. These features can be visualized in Figure 11. It is interesting that the feature filters look similar to two dimensional discrete cosine transformation (DCT) kernel functions. Based on this observation, it is clear that 2D spacial frequency information has been learned by the network as good discriminative features to distinguish the texture-like lung image patches.

CONCLUSION

A convolutional neural system (CNN) is a class of deep, feed forward systems, made out of at least one convolutional layers with completely associated layers to finish everything. This paper is the discussion about how the CNN algorithm is booming in every field finding its application. FPGA based applications of CNN as Stream Processor for embedded real time vision and Angel Eye, Memory-Centric and Embedded Streaming Deep Neural Networks accelerator designs, Real-Time Automation applications in Human Activity Recognition using Mobile Sensors and Hardware Design Automation, Image Processing applications in Medical Image Classification and Embedded facial image processing are the main fields of applications discussed here.

REFERENCES

1. Cl´ement Farabet, Cyril Poulet and Yann LeCun, "An FPGA-Based Stream Processor for Embedded Real-Time Vision with Convolutional Networks" , 12th international conference on compute vision workshops, 2009, pp: 878 - 885 .
2. Kaiyuan Guo, Lingzhi Sui, Jiantao Qiu, Jincheng Yu, Junbin Wang, Song Yao,Song Han, Yu Wang, Huazhong Yang, Senior Member," Angel-Eye: A Complete Design Flow for Mapping CNN onto Embedded FPGA", IEE Transactions on computer aided design of IC and systems, Volume: PP, Issue: 99,pp 1-1,2017.
3. Maurice Peemen, Arnaud A. A. Setio, Bart Mesman and Henk Corporaal," Memory-Centric Accelerator Design for Convolutional Neural Networks", 31st international Conference on Computer Design, 2013,pp: 13 – 19
4. Aysegul Dundar, Jonghoon Jin, Berin Martini, and Eugenio Culurciello, "Embedded Streaming Deep Neural Networks Accelerator With Applications", IEEE Transactions on Neural Networks and learning systems, Volume: 28, Issue: 7, 2017, pp: 1572 – 1583
5. Andrea Solazzo, Emanuele Del Sozzo, Irene De Rose, Matteo De Silvestri,Gianluca C. Durelli, Marco D. Santambrogio," Hardware Design Automation of Convolutional Neural Networks," IEEE computer society annual symposium on VLSI, 2016,Pages: 224 – 229
6. Ming Zeng, Le T. Nguyen, Bo Yu, Ole J. Mengshoel, Jiang Zhu, Pang

Wu, Joy Zhang, “Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors”,6th international conference on mobile computing applications and services, 2014, Pages: 197 – 205

7. Franck Mamalet, S´ebastien Roux, Christophe Garcia,” Embedded facial image processing with Convolutional Neural Networks”,International symposium on circuits and systems, 2010,pp: 261 – 264

8. Qing Li, Weidong Cai, Xiaogang Wang, Yun Zhou, David Dagan Feng∗ and Mei Chen, “Medical Image Classification with Convolutional Neural Network”,13th Internatiomal Conference on ICARCV, 2014.pp: 844 – 848.