DESIGN AND IMPLEMENTATION OF FUSIONET - A HYBRID MODEL TOWARDS IMAGE CLASSIFICATION

A PROJECT REPORT

Submitted by

MOLOKWU REGINALD CHUKWUKA (2016 224 001)

Under the guidance of

Dr. OGOCHUKWU OKEKE

(Doctorate, Department of Computer Science)

in partial fulfillment for the award of the degree

of

BACHELOR OF SCIENCE in COMPUTER SCIENCE

Submitted to

DEPARTMENT OF COMPUTER SCIENCE,

FACULTY OF PHYSICAL SCIENCE,

CHUKWUEMEKA ODUMEGWU OJUKWU UNIVERSITY, ULI

FEBRUARY 2021

CERTIFICATION

Certified that this seminar report titled: “**DESIGN AND IMPLEMENTATION OF FUSIONET - A HYBRID MODEL TOWARDS IMAGE CLASSIFICATION** ” is the bonafide work of “**MOLOKWU REGINALD CHUKWUKA (2016 224 001)** , who carried out the project work under my supervision. I further certify that, to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this candidate or any other candidate.

MOLOKWU REGINALD CHUKWUKA	Dr. OGOCHUKWU OKEKE
(2016 224 001)	**Supervisor**
Department of Computer Science	Department of Computer Science

Signature of the Student	Signature of Supervisor

APPROVAL PAGE

This is to approve that this project work titled DESIGN AND IMPLEMENTATION OF FUSIONET - A HYBRID MODEL TOWARDS IMAGE CLASSIFICATION written by MOLOKWU REGINALD CHUKWUKA Registration Number 2016 224 001 has been approved by the department of computer science Chukwuemeka Odumegwu Ojukwu University. In partial fulfillment of a bachelor of science degree.

______________________________	______________________________
Dr. Ogochukwu Okeke Supervisor	Date
______________________________	______________________________
Dr. Ogochukwu Okeke Head of Department	Date
______________________________	______________________________
Dr External Examiner	Date

ABSTRACT

Image classification, a topic of pattern recognition in Computer Vision (CV), is an approach of classification based on con-textual information in images. Contextual here means this approach is focusing on the relationship of the nearby pixels also called neighborhood. An open topic of research in CV is to devise an effective means of transferring human's informal knowledge into computers, such that computers can also perceive their environment. However, the occurrence of object with respect to image representation is usually associated with various features of variation causing noise in the image representation. Hence, it tends to be very difficult to actually disentangle these abstract factors of influence from the principal object. In this project, we have proposed a hybrid model: FUSIONET, which has been modelled for studying and extracting meaningful facts from images. Our proposition combines 2 distinct stack of convolution operation (3 x 3 and 1 x 1 respectively). Successively, these relatively low-feature maps from the above operation are fed as input to a downstream classifier for classification of the image in question

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my supervisor, Prof. (Mrs.) Ogochukwu Okeke, for the supervision, guidance, patience, and motivation. Her guidance has always been helpful during the course of my research work. I could not have imagined having a better supervisor.

Furthermore, I sincerely do appreciate the effort and commitment of the Head of Department and entire staff of the Department of Computer Science for their insightful comments, inspiration, criticisms, and encouragement; this has challenged me and motivated me thus far. You all provided me an opportunity to tap from your immense wealth of knowledge.

Also, I thank my fellow course mates (Nnamdi, Charles, and Ernest) for the inciting and stimulating discussions; and for the sleepless nights we were working together to meet deadlines in the last couple of years.

Lastly, but not the least, I would like to thank my family: my parents, my brothers especially Bonaventure C. Molokwu, and sisters for supporting me spiritually throughout the course of my study and research in school.

DEDICATION

This report is dedicated to GOD, my refuge and stronghold; as well to my benign and inspirational parents, Prof. and Mrs. C. C. MOLOKWU.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AI	Artificial Intelligence
CV	Computer Vision
GPUs	Graphics Processing Unit
CPUs	Central Processing Unit
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Networks
MLP	Multi-layer Perceptron
ANN	Artificial Nueral Network

CHAPTER 1

INTRODUCTION

1.1 Background of Study

According to Goodfellow et al. (2017), Deep Learning is an approach to machine learning that has drawn heavily on our knowledge of the human brain, statistics and applied mathematics as it developed over the past several decades, and this approach aims at allowing computers to learn from experience and understand the world in terms of a hierarchy of concepts, with each concept defined in terms of its relation to simpler concepts. Thus by gathering knowledge from experience, this approach avoids the need for human operators to formally specify all of the knowledge that the computer needs. The hierarchy of concepts allows the computer to learn complicated concepts by building them out of simpler ones. Therefore, if we decide to draw a graph showing how these concepts are built on top of each other; the graph is deep, with many layers. For this reason, the name Deep Learning (Deep Learning (DL)) was coined out.

Also, according to Deng and Yu (2014), they defined Deep Learning as a class of machine learning techniques which use many layers of information processing stages in hierarchical supervised architectures to exploit and resolve unsupervised feature learning as well as pattern analysis/classification. The essence of deep learning is to compute hierarchical features or representations of the observational data, where the higher-level features or factors are defined from lower-level ones.

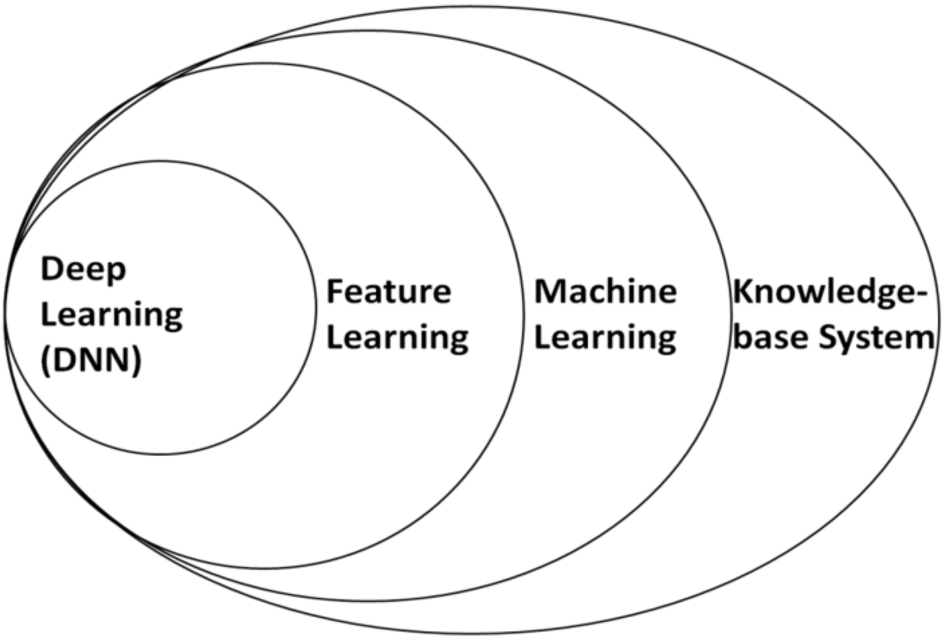

Figure 1.1: Deep Learning with respect to Artificial Intelligence evolution

Furthermore, several algorithms and techniques have been devised to train and induce learning in Deep Neural Networks (DNN). In this regard, these algorithms and/or techniques can be categorized or classified into three broad classes, namely:

(i) Supervised Learning;

(ii) Unsupervised Learning;

(iii) Reinforcement Learning.

1.1.1 Supervised Learning

This is a dependent training or learning process in neural networks which is carried out under the supervision of a guide known as ground truth (T) or labels. This training process begins with the input vector (X) which is fed into the neural network; and in turn, it produces an output vector (Y). Thus, the yielded output vector (Y) is compared with the ground-truth vector (T); and an error signal (S) is generated. The error signal $(S = T - Y)$ is basically the difference between the yielded output and the true/legitimate answer (usually a label). Then on the basis of this error signal (S), the weights associated with each neuron in the neural network are repeatedly adjusted until the yielded output is adequately matched with the ground truth.

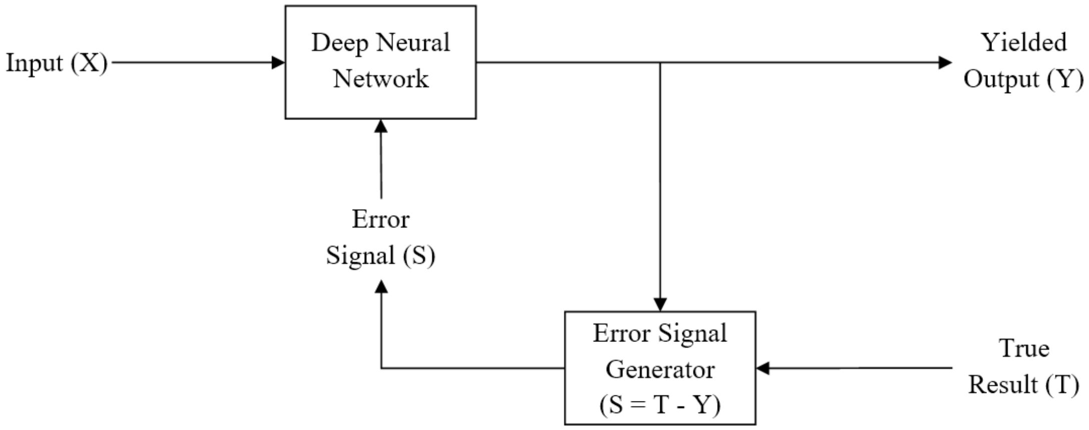

Figure 1.2: Architecture of a Supervised Learning model

1.1.2 Unsupervised Learning

This is an independent training or learning process in neural networks which is carried out without the supervision of guides/labels. This training process begins with gathering together input vectors of like identities to form clusters. Thus, when a new input vector is fed into this learning algorithm; the neural network yields an output indicating the class to which the input vector belongs to or is associated with. Also, there is no feedback from the environment with respect to what the ground truth is. Hence, in this type of learning, the neural network must self-discover the patterns and features of the input data as well as the relationships existing between the input data and the output.

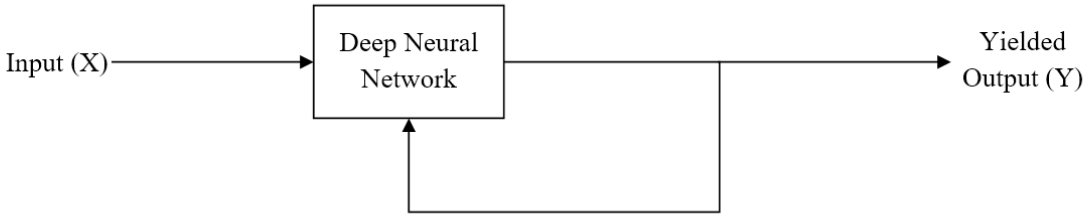

Figure 1.3: Architecture of a Unsupervised Learning model

1.1.3 Reinforcement Learning

Basically, as the name implies, this type of learning is used to reinforce or strengthen the neural network over some pivotal or focal information. This learning process is similar to supervised learning; however, we might have very less information. Thus, in reinforcement learning, the neural network model is exposed to the environment and allowed to execute tasks or solve problems. Thereafter, its actions are evaluated based on a performance measure. This performance measure can either be an utility/fitness function for positive actions; or a cost/loss function for negative actions. So if the network model executes a task rightly, it gets rewarded based on the utility/fitness function. On the other hand, if the model executes a task wrongly, it gets penalized based on the cost/loss function. Cumulatively, the neural network model over time learns what best actions to execute in future in a bid to acquire maximum rewards. This process is partially similar to the supervised learning model.

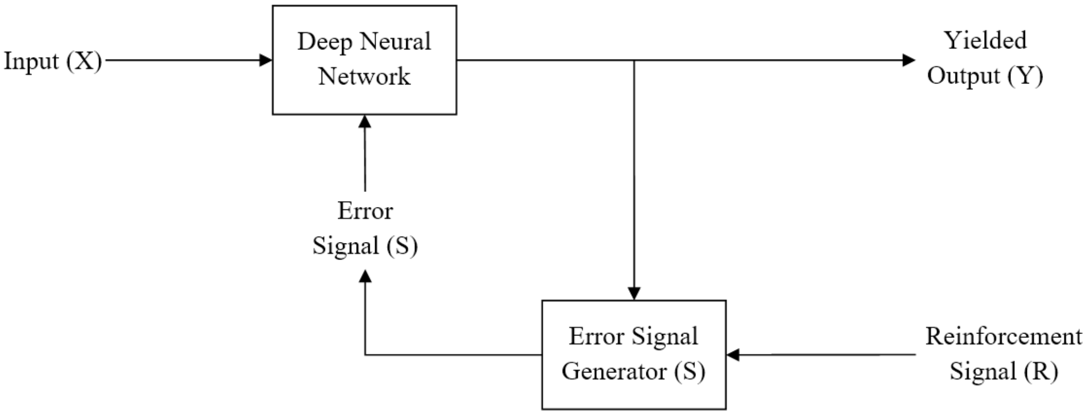

Figure 1.4: Architecture of a Unsupervised Learning model

1.2 Statement of Problem

A major hassle of Artificial Intelligence (AI) is to devise effective and efficient means of transferring humans' informal knowledge (like sense of image recognition, sense of speech, etc.) into machines and computers such that these machines can act and behave exactly like humans. However, the occurrence of objects with respect to image representations in real world is usually associated with various features of variation or factors of influence which constitute distractions or noise in the image representations. Hence, it tends to be very difficult to actually disentangle these abstract factors of influence from the principal object or observed entity. Thus, an effective Convolutional Neural Network (CNN) model should be able to identify and focus on the principal object we aim to observe; and disregard the associated distractions (features of variation). To that effect, these remain open problems and challenges to CNN and modern AI.

1.3 Aims and Objective of the Study

The goals of this study are:

1. Proposition of a DL-based and hybrid model, FUSIONET, modelled for image prediction and classification problems in image analysis.

2. Extensive bench-marking results which are centered on classic objective functions used for standard classifiers.

3. Comparative analyses, between FUSIONET and state-of-the-art methodologies, against standard real-world image dataset.

1.4 Scope of the Study

This study is limited to image recognition using CNN with respect to the research domain of CV. Our baselines (datasets) for benchmarking the performance of our proposed model are as enumerated in Table (4.2).

1.5 Significance of the research work

The findings of this study will be most appealing to researchers and students in the domains of Computer Vision, Deep Learning, and Machine Learning. Optimistically, my work summarized herein will substantiate and advance recent applications of Deep Learning via Convolutional Neural Networks.

With the increase in computer application, there is a need to strengthen the scope of computer vision, taking note that smart computer application must be able to to correctly and to certain degree be better at mimicking the human senses with accuracy and speed as a factor of the undersigned. The greater demands for efficient image learning algorithms justifies the need for more effective research approaches.

More also, the application of image classification especially in facial recognition for university examinations, attendance system, and for mobile portable device security and security at large, should benefit from the fruits of this study.

1.6 Definition of Terms

- **ARTIFICIAL INTELLIGENCE:** Computer science defines AI as “a system’s ability to correctly interpret external data, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation.”
- **ARTIFICIAL NEURAL NETWORK:** Artificial neural networks (ANN) or connectionist systems are computing systems that are inspired by, but not identical to, biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules.
- **ACTIVATION FUNCTION:** In artificial neural networks, the activat Tion function of a node defines the output of that node given an input or set of inputs.
- **BACKPROPAGATION:**Backpropagation Rumelhart et al. (1986) is an algorithm widely used in the training of feedforward neural networks for supervised learning; generalizations exist for other Artificial Nueral Network (ANN), and for functions generally. Backpropagation efficiently computes the gradient of the loss function with respect to the weights of the network for a single input-output example. This makes it feasible to use gradient methods for training multi-layer networks, updating weights to minimize loss
- **COMPUTER VISION:** Computer vision is an interdisciplinary field that deals with how

computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do

- **ACCURACY AND PRECISION:** In measurement of a training/testing set, accuracy refers to closeness of the measurements to a specific value, while precision refers to the closeness of the measurements to each other.

- **CONVOLUTIONAL NEURAL NETWORK:** The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

- **DEEP LEARNING:** Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces.

- **MULTI-LAYER PERCEPTRON:** Multi-layer Perceptron (MLP) is a class of feedforward artificial neural network. Learning occurs in the perceptron by changing connection weights after each piece of data is processed, based on the amount of error in the output compared to the expected result. This is an example of supervised learning, and is carried out through backpropagation, a generalization of the least mean squares algorithm

CHAPTER 2

LITERATURE REVIEW

2.1 Theoretical Review

Image classification is maybe the most significant piece of digital image analysis and computer vision Molokwu and Kobti (2019b). Image classification plays a pivotal role especially in face recognition, robot navigation, medical imaging and image search engines. With respect to advances in artificial intelligence (AI), real world (complex) images can be represented as vectors and analyzed by means of convolution neural network (CNN) operation. The exciting features of CNN is its ability to exploit spatial or temporal correlation in data LeCun et al. (1998).

In recent time, various improvement geared towards CNN methodology have been proposed to make CNN scalable to large heterogeneous, complex and multi class problems; this includes: modification of processing units Krizhevsky et al. (2012), He et al. (2016), Zeiler and Fergus (2014), Szegedy et al. (2016), parameter and hyper parameters Yoo (2019), Krishnakumari et al. (2020), Rawat and Wang (2019), Nguyen et al. (2019), optimization strategies Ismail et al. (2019), Ozcan and Basturk (2020), Dashdorj and Song (2019), Li et al. (2020) and design patterns Lu et al. (2020), Agarap (2017), Song et al. (2019), Suganuma et al. (2017).

Molokwu (2019) A significant hassle in computer vision is to model an effective and efficient algorithm of transferring humans' informal knowledge into machines and computers such that these machines/computers can act and behave exactly like humans. However, the occurrence of objects concerning image representations in the real-world is usually associated with various features of variation or factors of influence that constitute distractions or noise in the image representations. Hence, it tends to be very difficult to disentangle these abstract factors of influence from the principal object or observed entity. To that effect, these remain open problems, and challenges to image classification, computer vision and machine learning. Herein our proposed methodology is based on an iterative learning approach which is targeted at solving the problems of image classification by combining 2 convolution operation models in parallel. primarily, learning in FUSIONET is induced via supervised training and FUSIONET is capable of learning the non-linear distributed features enmeshed in an image vector.

2.2 Summary of Related works and Knowledge gap

As seen in the previous section, the available works have employed deeper and complex algorithms that are computationally expensive, they require weeks of training on complex Graphics Processing Unit (GPUs) and do not scale up efficiently in generalization and accuracy when applied to real word application. Hence,the novelty of our research contribution are stated below:

1. Proposition of a DL-based and hybrid model, FUSIONET, modelled for image prediction and classification problems in image analysis.
2. Extensive bench-marking results which are centered on classic objective functions used for standard classifiers.
3. Comparative analyses, between FUSIONET and state-of-the-art methodologies, against standard real-world image dataset.

Herein, we have evaluated FUSIONET across an array of state-of-the-art models and Deep Learning (DL)approaches which serve as our baselines, viz:

1. NAT (M3, M4): Neural Architectural Transfer Lu et al. (2020)
2. VGG: Very Deep Convolutional Networks for Large-Scale Image Recognition Simonyan and Zisserman (2014)
3. SOPCNN: Stochastic Optimization of Plain Convolutional Neural Networks with Simple methods Assiri (2020)
4. RESNET: Deep Residual Learning for Image Recognition He et al. (2016)
5. DENSENET-121: Densely Connected Convolutional Networks Huang et al. (2017)

CHAPTER 3

METHODOLOGY AND SYSTEM ANALYSIS

3.1 Methodology Adopted

The Methodology adopted for this project was Supervised learning. Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. Russell and Norvig (2002) It infers a function from labeled training data consisting of a set of training examples. Mohri et al. (2018) In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.

3.1.1 Algorithm Implemented

Algorithm used under supervised learning was the Artificial Neural Network ANN. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain.

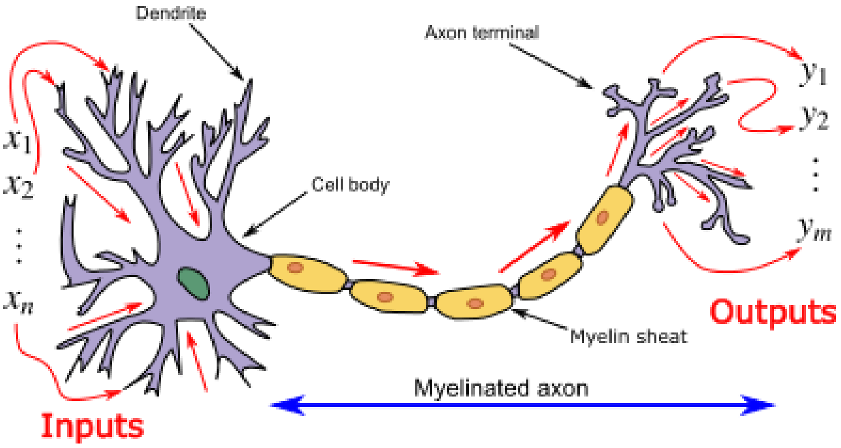

Figure 3.1: ANN inspiration - The human brain

As seen in figure 3.1, each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold. Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

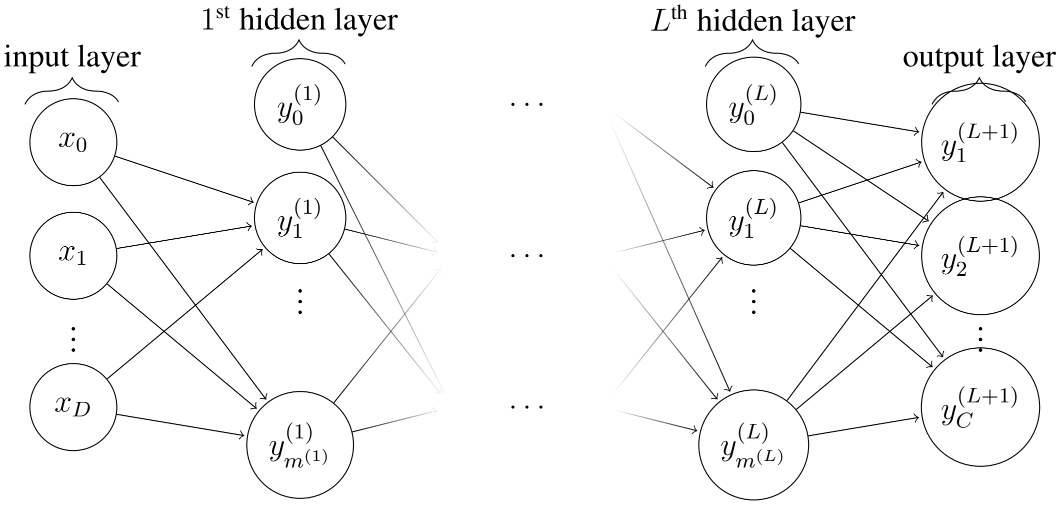

Figure 3.2: Network graph of a $(L+1)$-layer perceptron with D input units and C output units. The l^{th} hidden layer contains $m^{(l)}$ hidden units.

3.2 System Analysis

3.2.1 Analysis of the existing system

Image classification uses artificial intelligence technology to automatically identify objects, people, places and actions in images. One type of image classification algorithm is an image classifier. It takes an image (or part of an image) as an input and predicts what the image contains. The output is a class label, such as dog, cat or table. The algorithm needs to be trained to learn and distinguish between classes. This occurs by use of available ANN models and this models are very large in size because they have been trained on massive dataset, hence, scalability becomes a problem in that they require weeks of training on fast GPUs, sometimes months or years on training on Central Processing Unit (CPUs) for massive image datasets.

Weakness of the existing system

Traditional neural networks use a fully-connected architecture, as seen in 3.2, where every neuron in one layer connects to all the neurons in the next layer. A fully connected arctitecture is inefficient when it comes to processing image data:

1. For an average image with hundreds of pixels and three channels, a traditional neural network will generate millions of parameters, which can lead to overfitting.

2. The model would be very computationally intensive.

3. It may be difficult to interpret results, debug and tune the model to improve its performance.

3.2.2 Analysis of the proposed system

Unlike a fully connected neural network, in a Convolutional Neural Network (CNN) the neurons in one layer don't connect to all the neurons in the next layer. Rather, a convolutional neural network uses a three-dimensional structure, where each set of neurons analyzes a specific region or "feature" of the image. CNNs filters connections by proximity (pixels are only analyzed in relation to pixels nearby), making the training process computationally achievable. In a CNN each group of neurons focuses on one part of the image. For example, in a cat image, one group of neurons might identify the head, another the body, another the tail, etc. There may be several stages of segmentation in which the neural network image recognition algorithm analyzes smaller parts of the images, for example, within the head, the cat's nose, whiskers, ears, etc. The final output is a vector of probabilities, which predicts, for each feature in the image, how likely it is to belong to a class or category.

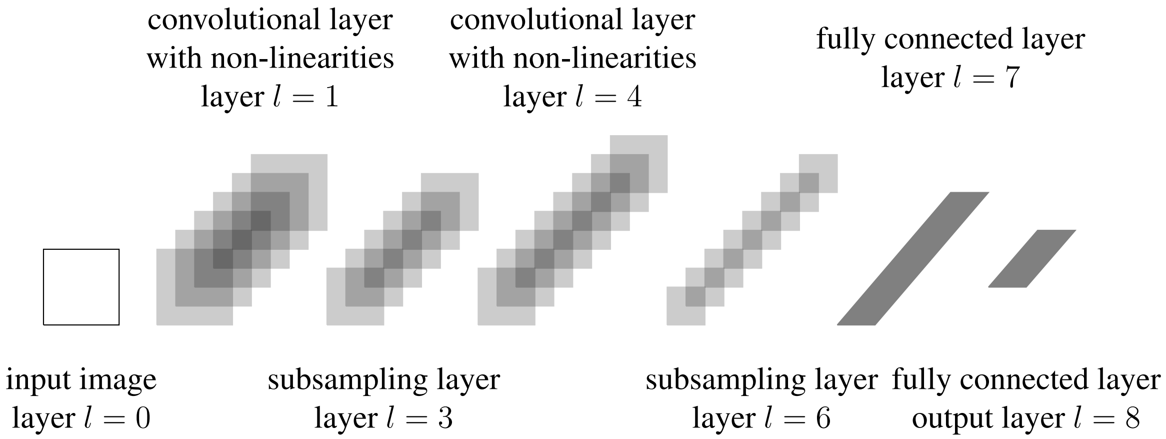

Figure 3.3: The architecture of the original convolutional neural network, as introduced by LeCun et al. (1989), alternates between convolutional layers including hyperbolic tangent non-linearities and subsampling layers. In this illustration, the convolutional layers already include non-linearities and, thus, a convolutional layer actually represents two layers. The feature maps of the final subsampling layer are then fed into the actual classifier consisting of an arbitrary number of fully connected layers. The output layer usually uses softmax activation functions.

Our proposed model (FUSIONET)

Fig 3.4 illustrates the architecture of our proposition, FUSIONET

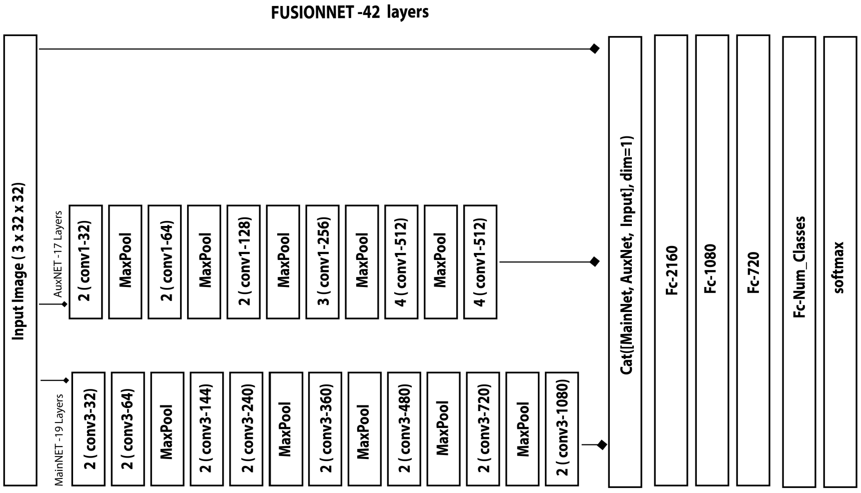

Figure 3.4: Proposed architectural framework of FUSIONET.

CHAPTER 4

SYSTEM DESIGN AND IMPLEMENTATION

4.1 Definition of problem

DEFINITION 1 Image Classification: Hashmi et al. (2020) Image classification is referred as a process of directly mapping floating integers to symbols

$$f(x) : x \Rightarrow \Delta; x \in R^n, \Delta = \{c1, c2, ..., c_L\} \tag{4.1}$$

Number of bands = n;

Number of classes = L

f(.) is a function that assigns a pixel vector x to a single class in the set of classes Δ

DEFINITION 2 Convolution Neural Network: Otherwise known as CNN or ConvNet LeCun et al. (1998) is a popular deep learning architecture and artificial neural network which is primarily used to extract and learn higher-order features in the dataset via convolutions, thus they are modeled based on the working principles of the visual cortex present in humans. CNNs are well adapted for learning features of image input due to its significant arrangement structure of neurons (or processing units) present in its respective processing layers.

DEFINITION 3 Convolution Operation: This is the fundamental operation of any Convolution Neural Network. It is responsible for extracting latent features and representations from the input image. Molokwu and Kobti (2019a), Molokwu and Kobti (2019b) A convolution is defined as a mathematical operation rule for merging two sets of data.

$$A_{xy} = (K * I)_{xy} = \sum_{0}^{i} \sum_{0}^{j} K_{ij}.I_{x+i}, y_{+j} \tag{4.2}$$

Formally, the above equation 2. represents a 2-dimensional convolution operation. Such that: A_{xy} represents a cell/matrix position in the Activation Map (or Feature Map); K_{ij} represents a cell/matrix position in the Kernel (or Filter or Feature Detector); and $I_{(x+i,y+j})$ represents a cell/matrix position in the Input matrix. As the index, K_{ij}, of the kernel slides from left-to-right and top-to-bottom; the index, $I_{(x+i,y+j})$, of the input matrix increases respectively - from left-to-right and top-to-bottom. Each resultant, A_{xy}, which is a convolution of the respective kernel and input indices is used to populate the activation/feature map

4.2 Proposed Methodology

Our proposition, FUSIONET, is comprised of (2) distinct ConvNet models *(MainNet, AuxNet)* and (1) classification layer.

4.2.1 MainNET

The MainNET is a stack of (19) layers of small 3 x 3 2D Convolution block. Sequel to the 2D Convolution block is Batch Normalization; a technique to provide any layer in a neural network with inputs that are zero mean/unit variance.
In this regard, Let B denote a mini-batch of size m training set, The empirical mean and variance of B is ascribed as

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i, and\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \tag{4.3}$$

The non-linearity activation function is a Parametric rectified linear unit (PReLU) which presents non-linearity after the batch normalization in the form of:.

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ \\ ax & \text{otherwise.} \end{cases} \tag{4.4}$$

The pooling function hereafter goes about as a specialist answerable for decreasing the data width of each activation map while holding its fundamental properties. subsequently, the Max-Pooling employed here is characterized with the end goal that the resultant pooled (or down-sampled) feature map is produced by means of: $p_i \in P = h(r_i \in R) = maxPool(R)$.

4.2.2 AuxNET

The AuxNet uses a (17) layers of small 1 x 1 Convolution block otherwise recognized as a linear transformation of the input, followed by Batch-Normalization, a non linear activation function (PRELU) and pooling (MaxPool).

4.2.3 Classifier

This is the last layer of our proposed FUSIONET design, and it succeeds the AuxNET layers, with (5) stack of Multi-layer perception (MLP). Classification is commonly the last phase of the FC vision framework, where the nearness and nonappearance of features is utilized to decide. The classifier matches distinguished examples against learned examples to distinguish the class of the input and make a score to show certainty. The pooled feature maps, created by both the MainNET and AuxNET which contains high level features removed from the constituent images in the dataset are concatenated alongside the input (x). Thus, the the classification uses these extricated "high level features" for recognizing images, in view of the individual classes. In this regard, LeCun et al. (1989) a MLP function is denoted as a mathematical function, f_c

that zips some set of input values, *P*, to their respective output labels, Y . In other words, $Y = f_c(P, \theta)$ and θ denotes a set of parameters.The MLP function models the estimations of θ that will bring about the best score, Y , approximation for the input set, P. The MLP classifier output is a likelihood dissemination which shows the probability of an image belonging to a particular class in the dataset.

4.2.4 Algorithm

Table 4.1: Proposed Image Classification Algorithm

```
**Input:** $\{V, E, \mathbb{Y}_{gTruth}\} \equiv$ {Images, Labels, Ground-Truth Entities}
**Output:** $\{\mathbb{Y}_{pred}\} \equiv$ {Predicted Entities}
**Preprocessing:**
$V \leftarrow$ Data *Augmentation* {(Rotation, Fliping, Cropping)}
$f_c, W \leftarrow$ *Initialize* {Construct classifier model, Weights}
**Training:**
$i \leftarrow 0$
**while** $i < 200$ **do**
    $out \leftarrow f_t \in F = (K \cdot I)_t;$
    $out \leftarrow p_t \in out = h(R) = maxPool_(out);$
    $out \leftarrow f_t \in out = (K \cdot I)_t;$
    $out \leftarrow p_t \in out = h(R) = maxPool_(out);$
    $f_c|\theta : out \rightarrow \mathbb{Y}_{gTruth};$
**return:**
$\{\mathbb{Y}_{pred} = f_c(\mathbb{Y}_{gTruth}, \theta)$ =0
```

4.3 Datasets

With regard to Image classification herein, five (6) real-world benchmark image classification data sets were employed for experimentation and evaluation, ranging from large scale to medium and small datasets viz: CINIC-10 Darlow et al. (2018), (CINIC-10, CIFAR-100) Krizhevsky et al. (2009), SVHN Netzer et al. (2011), FoodNET-20 Molokwu (2019), STL-10 Coates et al. (2011). See Table 4.2 and 4.3

Dataset	Type	Training set	Testing set	Classes
CINIC-10	Multi-class	120,000	90,000	10
CIFAR-10		50,000	10,000	10
CIFAR-100		50,000	10,000	100
SVHN		73,000	60,000	10
FoodNET-20		1,500	900	20
STL-10		5,000	8,000	10

Table 4.2: Benchmark Dataset for Evaluation.

Table 4.3: Configuration of experimentation hyperparameters.

Batch Size: 128	Optimizer: SGD (Momentum=0.99)
Activation: PReLU	Epochs: 1000
Dropout: 0.5	Learning Rate: 0.003
Learning Decay: after 300 epochs	weight decay: L2 penalty
Stride: 1	L2 Multiplier: 5e-4
Maxpooling: (2, 2)	Number of Parameters: 64M

4.4 System Implementation

4.4.1 Hardware requirements

Building a deep learning system can be intimidating and time-consuming. In this regard, the minimum requirements in building this architectural framework are:

1. GRAPHICS CARD:- Nvidia CUDA enabled GTX 1650 or 1660TI
2. CPU:- i5 (9th or 10th Generation)
3. RAM:- 16GB

4.4.2 Software requirements

There are no extra software required for image classification using Convolutional Neural Network CNN. All you need are:

1. Python IDE (Jupyter or Spyder Preferably)
2. Deep Learning Toolkit (Keras, Tensorflow or Pytorch preferably)

4.5 Data Pre-processing and Training

All benchmark image datasets ought to be cleaned for efficient classification and reduction in training time.

On training, the inputs (x) to our ConvNets are fixed-size 32 × 32 RGB image. Preprocessing employed includes manually correction of wrong labelled inputs, Normalization, Flipping of random images horizontally and vertically, randomly cropping the centers of some images, erasing some parts of the images, altering the colors of random images and random perspective.

4.6 Experiments and Results

FUSIONET's experimentation setup was tuned in accordance with the hyperparameters shown in Table 4.3. Categorical Cross Entropy was employed as the cost/loss function; while the fitness was measured with reference to accuracy at *Top1, Top5 and Flops*. Moreover, the accuracy have been computed against each benchmark data set with regard to the constituent classes (or categories) present in each data set.

With regard to the image classification tasks herein, the performance of our FUSIONET model while benchmarking against five(5) popular baselines (NAT-M4, NAT-M3, SOPCNN, ResNeXt29x64d); and when evaluated against the validation/test samples for the benchmark data sets are as documented in Table 4.4, 4.5, 4.9, 4.7, 4.8

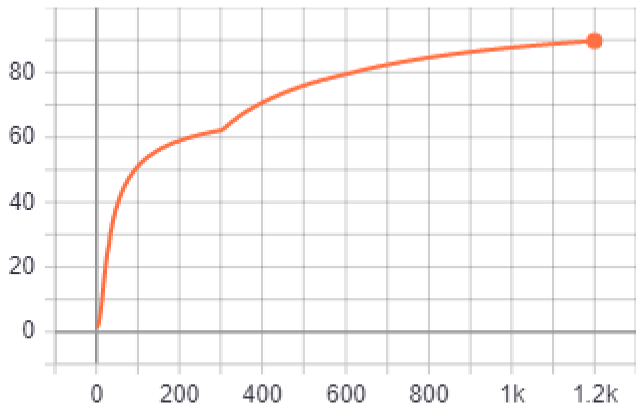

Figure 4.1: Epochs Vs Accuracy on CIFAR-100 dataset after 1200 Epochs

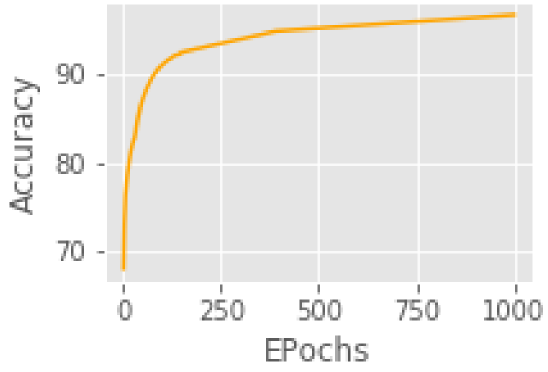

Figure 4.2: Epochs Vs Accuracy on CINIC-10 dataset after 1000 Epochs

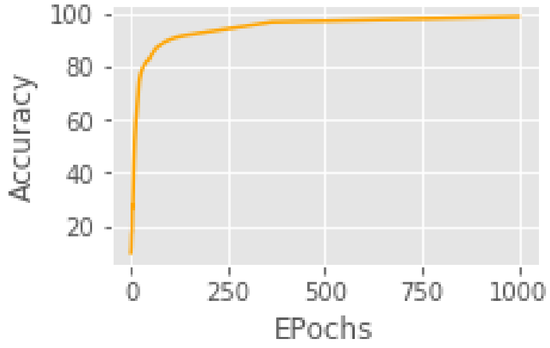

Figure 4.3: Epochs Vs Accuracy on CIFAR-10 dataset after 1000 Epochs

Table 4.4: Image classification over CINIC-10 data set. Results are based on the set apart validation samples

		CINIC-10		
Model	FLOPS	Top 1 accuracy	Top 5 accuracy	Error rate
NAT-M4	710M	94.80%	99.30%	5.20%
NAT-M3	**501M**	94.30%	98.60%	5.70%
ResNeXt29x64d	600M	91.45%	98.40%	8.55%
VGG-16	690M	87.77%	97.55%	12.23%
DenseNET-121	**500M**	91.26%	98.53%	8.74%
ResNET-18	505M	90.27%	98.01%	9.73%
FUSIONET	569M	**96.84%**	**99.88%**	**3.16%**

Table 4.4. Shows our FUSIONET achieving state-of-the-art on the CINIC-10 dataset for

top-1 and top-5 accuracy. Improving NAT-M4 by 2.04% for top-1 and 0.58% for top-5. floating point operations per second (FLOPS) was benched at 569M with regards to NAT-M4 of 710M.

Table 4.5: Image classification over CIFAR-10 data set. Results are based on the set apart validation samples

CIFAR-10				
Model	FLOPS	Top 1 accuracy	Top 5 accuracy	Error rate
NAT-M4	468M	98.40%	99.60%	1.60%
NAT-M3	392M	97.20%	98.30%	2.80%
ResNeXt29x64d	488M	96.71%	98.40%	3.29%
SOPCNN	440M	94.29%	98.00%	5.71%
FUSIONET	**386M**	**98.54%**	**99.82%**	**1.46%**

Table 4.6: Image classification over STL-10 data set. Results are based on the set apart validation samples

STL-10				
Model	FLOPS	Top 1 accuracy	Top 5 accuracy	Error rate
NAT-M4	573M	92.61%	**98.55%**	7.39
NAT-M3	**436M**	97.80%	98.48%	2.20%
ResNeXt29x64d	476M	80.61%	96.40%	19.39%
SOPCNN	424M	88.08%	97.02%	11.92%
FUSIONET	440M	**97.90%**	98.76%	**2.10%**%

Table 4.7: Image classification over SVHN data set. Results are based on the set apart validation samples

SVHN				
Model	FLOPS	Top 1 accuracy	Top 5 accuracy	Error rate
NAT-M4	610M	**98.51%**	**99.67%**	**1.49%**
NAT-M3	520	97.80%	99.24%	2.20%
ResNeXt29x64d	533M	98.50%	99.48%	1.50%
SOPCNN	512	98.50%	99.55%	1.50%
FUSIONET	**494M**	97.63%	99.31%	2.37%

Table 4.8: Image classification over CIFAR-100 data set. Results are based on the set apart validation samples

CIFAR-100				
Model	FLOPS	Top 1 accuracy	Top 5 accuracy	Error rate
NAT-M4	796M	88.30%	**95.71%**	11.70%
NAT-M3	492M	87.70%	95.55%	12.30%
ResNeXt29x64d	491	83.44%	94.48%	16.56%
SOPCNN	501	72.96%	90.22%	27.04%
FUSIONET	**470M**	**89.72%**	**96.86%**	**10.28%**

Table 4.9: Image classification over FoodNET-20 data set. Results are based on the set apart validation samples

FoodNET-20				
Model	FLOPS	Top 1 accuracy	Top 5 accuracy	Error rate
NAT-M4	443M	82.61%	94.55%	17.39
NAT-M3	**366M**	87.80%	94.48%	12.20%
ResNeXt29x64d	416M	70.61%	88.40%	29.39%
SOPCNN	394M	78.08%	87.02%	21.92%
FUSIONET	370M	**91.90%**	**98.76%**	**8.10%%**

4.6.1 Performance evaluation

In this manner, we have highlighted the model which performed best (considering *Top-1, Top-5* precision and FLOPS where conceivable), for each classification task using a bold font. We utilized a point-based reviewing standard to discover the fittest model for each image classification task. The model with the most noteworthy combined point altogether wins the fittest model for each image classification task. Accordingly, as can be seen from our tabular results, FUSIONET is at the top with the highest fitness points; and this superb performance can be linked to two (2) primary factors,namely:

1. The combination of 2 distinct convolution stack (MainNET and AuxNET) in FUSIONET conception model

2. The top notch data pre-processing techniques employed herein with respect to the benchmark datasets. We ensured that all images used for training were in the standard format and classes.

4.6.2 Limitations of the system

The benchmark models evaluated herein were implemented using their default parameters except for CIFAR-100. To prepare the CIFAR-100 experiment, we changed certain parameters including data augmentation techniques. We decreased the batchsize to 64 and expanded the learning rate and the training epochs. Every other hyperparameters were left unaltered throughout the training process. In summary, FUSIONET's remarkable performance with respect to our benchmarking results can be attributed to the presence of 2 convolution stack running in parallel.

CHAPTER 5

APPLICATION AND CONCLUSION

5.1 Summary

Machine learning with deep layer artificial neural networks (DL)has been gaining momentum over last decades. The successful results gradually propagate into our daily live. Convolutional neural networks (CNN) is a special architecture of artificial neural networks, proposed by LeCun et al. (1989). CNN uses some features of the visual cortex. One of the most popular uses of this architecture is image classification. Image classification refers to a process in computer vision that can classify an image according to its visual content. For example, an image classification algorithm may be designed to tell if an image contains a human figure or not. While detecting an object is trivial for humans, robust image classification is still a challenge in computer vision applications.

5.2 Conclusion

We propose a deep layer mathematical learning algorithm for real time image classification, which employs the use of convolutional neural network CNN. CNN is more contrast robust and overcomes the limitation of locality of multi-layer perceptron MLP. We combined the CNN with MLP classifiers in a novel approach to classify images in the dataset as seen in table 4.2

5.3 Recommendation

1. With the increase trend in image classification, Tertiary Schools in Nigeria should begin to adopt the scope of Image recognition for attendance management and exam imperson-

ation discovery.

2. The Federal Government of Nigeria, should invest in further research of the field of Machine learning, being a new field with little experts and broad scope.
3. Industries in Nigeria, should begin to adopt image classification algorithms for object labeling, recognition and detection
4. Deep layer Networks with automatic feature extraction have proven to be more efficient and time saving in the field of image classification, therefore should be widely used in image classification tasks rather than manual feature extraction.

5.3.1 Application Areas

Image classification is a distinct niche of computer vision that has seen various practical application. A ground-breaking application of image classification can be found in the field of stock photography and video. Stock sites give stages where picture takers and video makers can sell their content. contributors need an approach to label a lot of visual material, which is tedious and dreary. In a similar time, without appropriate catchphrase attribution, their content can't be indexed – and in this manner can't be found by purchasers.

Visual recognition on social media is already a fact. Facebook released its facial recognition app Moments, and has been using facial recognition for tagging people on users' photos for a while. Iris recognition is a widely used method for biometric identification. It's most common application is in border security checks, where a person's identity is verified by scanning their iris. The identification is conducted by analyzing the unique patterns in the colored part of the eye. In the last years, self-driving cars are the buzz in the auto industry and the tech alike. Autonomous vehicles are already being actively tested on U.S. roads as at the writing of this study. Forty-four companies are currently working on different versions of self-driving vehicles.

Besides the impressive number of image recognition applications in the consumer oriented market, it is already employed in important manufacturing and industrial processes. Teaching machines to recognize visuals, analyze them, and take decisions on the basis of the visual input holds stunning potential for production across the globe.

5.3.2 Future work

In conclusion, we aspire to expand FUSIONET scope to include more computer vision problems. Also we are sourcing for additional benchmarking models and real world (complex) datasets for exhaustive validation of FUSIONET.

REFERENCES

Agarap, A. F. (2017). "An architecture combining convolutional neural network (cnn) and support vector machine (svm) for image classification." *arXiv preprint arXiv:1712.03541*.

Assiri, Y. (2020). "Stochastic optimization of plain convolutional neural networks with simple methods." *arXiv preprint arXiv:2001.08856*.

Coates, A., Ng, A., and Lee, H. (2011). "An analysis of single-layer networks in unsupervised feature learning." *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. 215–223.

Darlow, L. N., Crowley, E. J., Antoniou, A., and Storkey, A. J. (2018). "Cinic-10 is not imagenet or cifar-10." *arXiv preprint arXiv:1810.03505*.

Dashdorj, Z. and Song, M. (2019). "An application of convolutional neural networks with salient features for relation classification." *BMC bioinformatics*, 20(10), 244.

Deng, L. M. and Yu, D. H. (2014). *Deep Learning: Methods and Applications*. Foundations and trends in signal processing. Now Publishers.

I. G. Goodfellow, Y. Bengio, and A. C. Courville, eds. (2017). *Deep Learning*. MIT Press, Cambridge, MA.

Hashmi, M. F., Kumar, A., and Keskar, A. G. (2020). "Subjective and objective assessment for variation of plant nitrogen content to air pollutants using machine intelligence: Subjective and objective assessment." *Fuzzy Expert Systems and Applications in Agricultural Diagnosis*, IGI Global, 83–108.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

Ismail, A., Ahmad, S. A., Soh, A. C., Hassan, K., and Harith, H. H. (2019). "Improving convolutional neural network (cnn) architecture (minivggnet) with batch normalization and learning rate decay factor for image classification." *International Journal of Integrated Engineering*, 11(4).

Krishnakumari, K., Sivasankar, E., and Radhakrishnan, S. (2020). "Hyperparameter tuning in convolutional neural networks for domain adaptation in sentiment classification (htcnn-dasc)." *Soft Computing*, 24(5), 3511–3527.

Krizhevsky, A., Hinton, G., et al. (2009). "Learning multiple layers of features from tiny images.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). “Imagenet classification with deep convolutional neural networks.” *Advances in neural information processing systems*. 1097–1105.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). “Backpropagation applied to handwritten zip code recognition.” *Neural computation*, 1(4), 541–551.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). “Gradient-based learning applied to document recognition.” *Proceedings of the IEEE*, 86(11), 2278–2324.

Li, H., Chaudhari, P., Yang, H., Lam, M., Ravichandran, A., Bhotika, R., and Soatto, S. (2020). “Rethinking the hyperparameters for fine-tuning.” *arXiv preprint arXiv:2002.11770*.

Lu, Z., Sreekumar, G., Goodman, E., Banzhaf, W., Deb, K., and Boddeti, V. N. (2020). “Neural architecture transfer.” *arXiv preprint arXiv:2005.05859*.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.

Molokwu, B. and Kobti, Z. (2019a). “Spatial event prediction via multivariate time series analysis of neighboring social units using deep neural networks. 1–8.

Molokwu, B., Shuvo, S. B., Kar, N. C., and Kobti, Z. (2020a). “Node classification and link prediction in social graphs using rlvecn.” *32nd International Conference on Scientific and Statistical Database Management*. 1–10.

Molokwu, B. C. and Kobti, Z. (2019b). “Event prediction in complex social graphs via feature learning of vertex embeddings.” *International Conference on Neural Information Processing*, Springer. 573–580.

Molokwu, B. C., Shuvo, S. B., Kar, N. C., and Kobti, Z. (2020b). “Node classification in complex social graphs via knowledge-graph embeddings and convolutional neural network.” *International Conference on Computational Science*, Springer. 183–198.

Molokwu, R. (2019). “An advanced movie recommender engine implemented in python.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). “Reading digits in natural images with unsupervised feature learning.

Nguyen, L. D., Gao, R., Lin, D., and Lin, Z. (2019). “Biomedical image classification based on a feature concatenation and ensemble of deep cnns.” *Journal of Ambient Intelligence and Humanized Computing*, 1–13.

Ozcan, T. and Basturk, A. (2020). “Performance improvement of pre-trained convolutional neural networks for action recognition.” *The Computer Journal*.

Rawat, W. and Wang, Z. (2019). “Hybrid stochastic ga-bayesian search for deep convolutional neural network model selection..” *J. UCS*, 25(6), 647–666.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). “Learning representations by back-propagating errors.” *nature*, 323(6088), 533–536.

Russell, S. and Norvig, P. (2002). “Artificial intelligence: a modern approach.

Simonyan, K. and Zisserman, A. (2014). “Very deep convolutional networks for large-scale image recognition.” *arXiv preprint arXiv:1409.1556*.

Song, T., Meng, F., Rodriguez-Paton, A., Li, P., Zheng, P., and Wang, X. (2019). “U-next: A novel convolution neural network with an aggregation u-net architecture for gallstone segmentation in ct images.” *IEEE Access*, 7, 166823–166832.

Suganuma, M., Shirakawa, S., and Nagao, T. (2017). “A genetic programming approach to designing convolutional neural network architectures.” *Proceedings of the genetic and evolutionary computation conference*. 497–504.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). “Rethinking the inception architecture for computer vision.” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–2826.

Yoo, Y. (2019). “Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches.” *Knowledge-Based Systems*, 178, 74–83.

Zeiler, M. D. and Fergus, R. (2014). “Visualizing and understanding convolutional networks.” *European conference on computer vision*, Springer. 818–833.