Exer-NN: CNN BASED HUMAN EXERCISE POSE CLASSIFICATION

BY

Md. WARID HASAN
ID: 161-15-7142

AND

JANNATUL FARDOSH NIMA
ID: 161-15-7381

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

MS. NISHAT SULTANA
Lecturer
Department of CSE
Daffodil International University

Co-Supervised By

MD. FERDOUSE AHMED FOYSAL
Lecturer
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

DECEMBER 2019

APPROVAL

This Project titled "**Exer-NN: Cnn Based Human Exercise Pose Classification**", submitted by Md. Warid Hasan, ID No: 161-15-7142 and Jannatul Fardosh Nima, ID No: 161-15-7381 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 07-12-2019.

BOARD OF EXAMINERS

Dr. Syed Akhter Hossain **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Md. Zahid Hasan **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Sadekur Rahman **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Dewan Md. Farid **External Examiner**
Associate Professor
Department of Computer Science and Engineering
United International University

DECLARATION

We hereby declare that this research has been done by us under the supervision of **Ms. Nishat Sultana, Lecturer, Department of Computer Science and Engineering and** co-supervision **of Md. Ferdouse Ahmed Foysal, Lecturer, Department of Computer Science and Engineering**, Faculty of Science and Information Technology, Daffodil International University. We also declare that neither this research nor any part of this research has been submitted elsewhere for the award of any degree.

Supervised by:

Nishat

Ms. Nishat Sultana

Lecturer

Department of CSE

Daffodil International University

Co- Supervised by:

Foysal

Md. Ferdouse Ahmed Foysal

Lecturer

Department of CSE

Daffodil International University

Submitted by:

Warid Hasan

Md. Warid Hasan
ID: 161-15-7142
Department of CSE
Daffodil International University

Jannatul fardosh

Jannatul Fardosh Nima
ID: 161-15-7381
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Ms. Nishat Sultana**, **Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of Deep Learning to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Syed Akhter Hossain**, and Head**,** Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

At last, again we want to thank all the good wishers, friends, family, seniors for all the help and inspirations. This research is a result of hard work and all those inspirations and assistance.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

To enjoy the glow of good health, you must exercise [Gene Tunney], because it helps us to feel happier, increase energy levels, reduce chronic disease and helps us to keep our brain and body refresh. Technology can make an important role to help aspects of our life in different portions. Today's computer vision technology uses deep learning algorithms that use the so-called convolutional neural networks (CNN), make sense of pictures. We can use the convolutional neural network (CNN) in deep learning to get state-of-art accuracy in different in various classification problems like as Image data, CIFAR-100, CIFAR-10, MINIST data sets. In this work, we propose a novel system to classify different types of human exercise pose detections automatic self-ruling decision making and predictive models using Convolutional neural networks (CNN). In earlier a lot of research has been conducted to pose detections in image classification problems, but our related tropic human exercise pose detection problem has few works on different data sets and different models with low accuracy. For strong architecture, we retrained the final layer of the CNN architecture, VGG16, MobileNet, Inception V3 for classification approach. We will create a new CNN model name ‘Exer-NN’ to successfully classify human exercise pose. Predicting among five different classes (bench press, bicep curl, squat, deadlift, treadmills). We proposed an average accuracy is 88% approximately that can be used for different purposes like tool kit assistance, helping management system automatically.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
Introduction

1.1 Introduction

In recent years, computer vision and artificial intelligence are widely used for human activity analysis like face detection, understand emotion, pose detection, etc. technology has applied in different ways. We can apply pose detections technology for exercise activities using computer vision applications. Now every country's people believed that exercise is very important for the physical and mental of our health, that's country could be developing or developed. Nowadays in our country this exercise zone increasing significantly, so we can make automatic decision making an application to apply in an exercise zone to helping management system and some other tool kit. We can use deep learning using a convolutional neural network (CNN) that read every pixel of an image and successfully able to extract feature from an image. There are different types of exercise, but we work only with 5 popular exercise pose detections and whose name is bench press, bicep curl, deadlift, squat, treadmills. Our main goal is to build a transfer learning model that can recognize a photo of different types of exercise poses that could apply computer vision applications. For that, firstly we have to train our Exer-NN model with a different type of image data, but we developed a new Exer-NN CNN model to train our images for better accuracy. And also train VGG16, InceptionV3, MobileNet models for comparison with our CNN model for more trustworthy. There are several parts to complete our task, for easy to understand we can divide our task into different following sections, such as in section 3 described our proposed methodology that’s included in an implementation of our model, data collection, data augmentation, data preprocessing, define test set for evaluating our models and train data set to train our model. In section 4 are described as performance evaluation, result discussion, comparison. And future work and conclusions and future works are mentions in section 5.

1.2 Motivation

Technology is an integral component of every human being's because with technology we can make easier our daily life. Now we can use artificial intelligence that able to make a decision like humans. When we focus on an object or something then in a short time we can able to make a clear sense of that object. Our brain is very strong and faster to make a clear sense, but it is not easy for us. Every time we processing a minimum of 60 images with high-resolution pixels to recognize that. When an object's light goes through our retina with neurons to our brain then we can see and understand about that object but take a little bit of time to give a result. To understand something we have to train our brain, like that if we want to make a machine that can see the object and identify it successfully. For that, we have to teach a machine to classify objects near humans. At that idea, we can build a model that can classify an image with a relation of the training task.

1.3 Rationale of the Study

Using artificial intelligence computers are becoming more and more human because there is no department we don't use computers. We can use artificial intelligence in anywhere like as internet, playground, home, office, factory, etc. For a helping management system, we can use it in a different way and in different places. By considering that's facts we can make a model that successfully helps a human exercise zone management system. Our proposed model makes a vital role in successfully recognized human exercise activities efficiently. For making a model we can use various machine learning models like deep learning using a convolutional neural network.

1.4 Research Questions

For the first time, we can't understand what is our first work? because it is very challenging for us to identify our right path. At that time we feeling some confusion questions.

- which programing language is perfect to implement our problems.
- Which types of image are performed very well.
- How people are benefited?
- Is it possible for a 90% correct classification rate?

- How easily we can use that?

1.5 Expected Outcome

We want to build a model that successfully helps guide decision making. That model works on human exercise pose detections accurately and can able to predict for helping decision making. In our project, we want to build a model using popular most recent machine learning techniques for the best performance. Our model perfectly identifies human exercise activity and give feedback. Our model works in indoor and outdoor perfectly. It's can able to predict 90% accurately for every input. Our model will be lightweight because it takes a little bit of time to give outputs for every input. Our model read-only image form camera, video, textfile, manually, etc. But the model's output can be a different way. To build our model we use Deep Learning of Convolutional neural network. Because the Convolutional neural network gives a good classification performance for image data-sets. There is some main target for the outcome of our project.

- Human exercise pose activity classified.
- A little bit of time to give output.
- It helps management systems by providing informations.
- It could able to perform indoor and outdoor.

1.6 Layout of the Report

Chapter one gave an introduction to the project with introduction, motivation, rational of study. We defined some research questions and provide some expected outcomes of our research project. In this part, we describe the whole report layout of our thesis.

In chapter two we will discuss background study, Manson some related works of our thesis, define some scope of the problems, and highlight some challenges.

In chapter three we will discuss our proposed method theoretically. But firstly we will mention some procedures like data collection, data augmentation, preparing, etc. We will provide some pictures and mathematical equations for clear convolutional neural network concepts. We will explain every layer of the model, test set, and training method. We will address the minimum requirements for this analysis at the end of chapter three.

Chapter three discusses the performance comparison of our Exer-NN models with some popular models such as inceptionV3, VGG16, and MobileNet. At last, for our model and for every model, we will complete the result discussion.

Chapter five discussed with the summary of the study, mentioned some future work and conclusion of our study. This chapter is responsible for demonstrating that the entire project report follows the recommendation. We mentioned some problems and limitations of our works, but the chapter is closed by shown some bright future scope are specified

CHAPTER 2

Background Study

2.1 Introduction

In this section, we discuss related works of our project, background information, some scope of the problems and challenges. That discussion is helpful for our project to solve some problems because we can find relevant information from our related works or literature reviews. In this section, we can identify the scope of the problems and challenges to how to increase the accuracy level.

2.2 Related Works

Before the widespread works on image classifications with different data sets or different algorithms. But till now, there are few image-based human exercise pose detection worked. Sadeka Haque proposed an ExNET model that able to classify image-based human exercise pose detections, and they got the best accuracy of 82.68% [1]. Terry Taewoong Um and they used [2] large scale wearable sensor data used to classify different types of human exercises with 92.14% classification accuracy. A still human exercise image we can classify and represent in many different ways, [3] in this work they represent their output using the Pictorial Structures model to human pose estimation. This pictorial Structures configured the human body structure with some Stiffness part in an efficient way, and generate pose prediction. Alexander Toshev and Christian Szegedy [4] are proposed a method that able to classification human pose estimation based on Deep Neural Networks. Main challenge of estimate human joint location, and visible with strong architecture. DNN- based regression gives a good performance for localization but CNN better for a classification task. In 2015 [5] Liang-Chieh Chen developed the DeepLab model that can be performed at a time classification and object detection task with computational efficiency and significantly advanced. DeepLad gives 71.6% accuracy for image segmentation, other hands this model can be used for any image classification task, and depth maps or videos. From another side for improving semantic segmentation [6], Siddhartha Chandra and Lasonas Kokkinos introduce the solution of a linear system, with a prediction technique with deep learning. Human [7], daily 5 physical activity (walking, Running, Cycling, Driving, Sports) data collected by the Philips NWS Activity Monitor and classified with Bayesian classification accuracy 72.3% and compare with Decision Tree algorithm.

2.3 Research Summary

First machine learning research is carried out with simple algorithms, but day by day increasing machine learning algorithm, complexity to solve the more and more complex issue, so we can say that machine learning increasing significantly for every portion of our life. From a few years ago image-based classification started using some machine learning techniques. Using Convolutional neural networks we can able to teaching machines to understand an image. In this image classification part, there are a lot of works in a different way. We get better knowledge from the background of machine learning, and related works. Now we can say there are some works on human exercise activity detection, but we make a new model that can be able to classify human exercise activity more efficiently and accurately.

2.4 Scope of the problem

For every step of our task, there is so much section for the problem. Once we start a new job then it is not possible to define problems, but during the research, we can understand that. After data collection, we have to manipulate them in any way before training. Programing syntax, model selection is the most challenging works for working.

2.5 Challenges

There are several challenges to complete our project. First, we have to collect data set form different platforms to train our model. We have to prepare data before going through the CNN model. The big challenge to build a new model with various layers. Define training set, avoid Overfitting, select a number of epochs and batch size are more challenging for us. There are most important for computer requirements because often every model creates 7 cores and more parameters. To train this huge number of parameters we need GPU, because using GPU need often half-hour, but without GPU we need 7-8 hours. It is a very challenging task. So we can say that our task is difficult, but it can be successfully completed.

CHAPTER 3

Research Methodology

3.1 Introduction

In chapter three, we going to fully describe working on our methodology. In this research, we apply convolutional neural network for image classification, because it is the most powerful and popular for deep learning. we are going to use our own created data set and model of the Convolutional neural network (CNN). we discuss training and test set, input, output, every convolutional layer, different types of technical issues, performance, etc. For the cleaning concept, we will give a proper example.

3.2 Research Subject and Instrumentation

The research subject makes a clear concept of our research area. In this section, we implement and design our model, collect perfect data, prepare them and train our model, performance discussion, and then apply our model to work. To complete our task we used windows platform. For implementing our task we use python programing language and many other packages like Keras, Tensorflow, numpy, cv2, seaborn, matplotlib, etc. We use python's another package mini-conda to complete all programming tasks on a jupyter notebook. We choose Python because of its free, simple syntax, foster fast testing for complex algorithms readability for machine learning applications.

3.3 Data Collection Procedure

In this study, a make new dataset is to train the proposed networks. Those images collected from various platforms like websites, Facebook, Instagram, Reddit, etc. All image is JPG format and resolution are not same. There are more than 2873 images with five categories like bench press, bicep curl, deadlift, squat, treadmills. This dataset was collected manually from the internet and publicly available information on it. Each class has more than 90 images. This dataset has a total of 2873 images, 2154 images for training and 20% of 719 for testing.

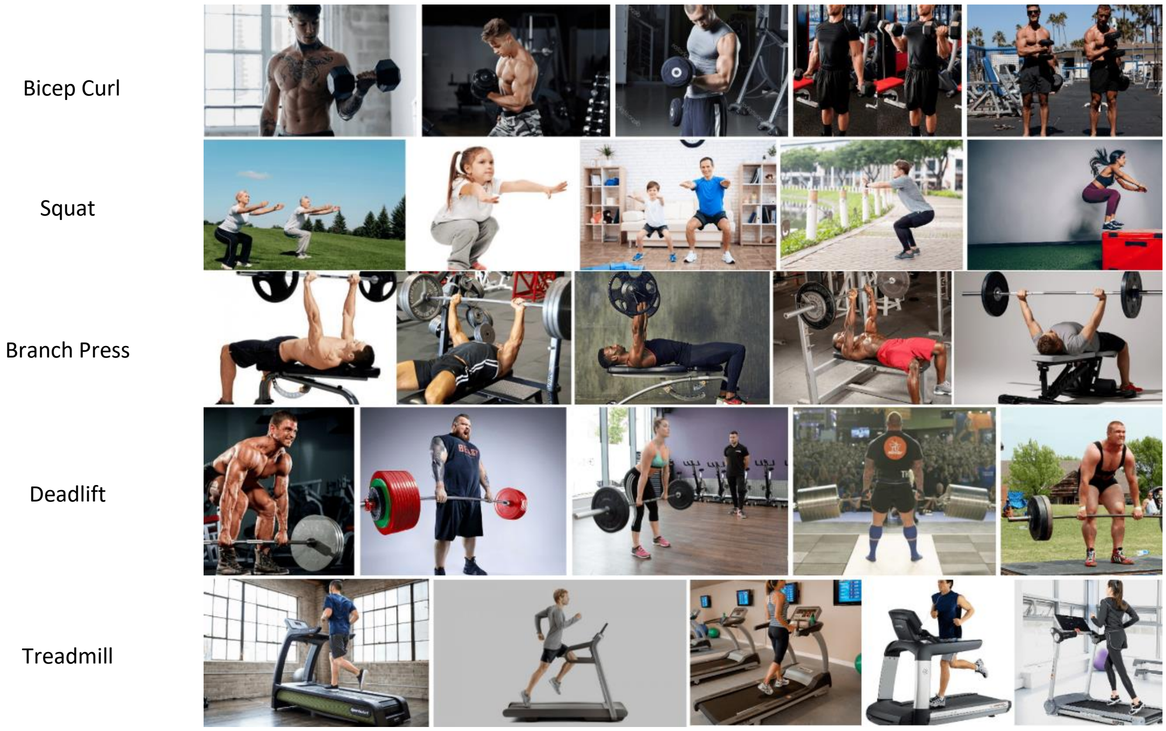

Fig 3.1: A small part of the data set

3.4 Data Processing

Our first challenge is to collect data and process them. When we collect our data set those all images resolution are not the same, then we need to prepare them for work. Data processing is the most important for training the model and get better accuracy, because it reduces the computational cost, reduces overfitting, etc.

3.5.1 Data Augmentation

A few training sets can result in overfitting [11] The number of new dataset samples was increased by using basic types of image increase to avoid overfitting [12] which we used to train our convolutional neural network model (CNN). Data augmentation is very important because it makes more effective of model performance and reduces classification loss. We prepared total data using in 3 different methods, these methods given below by a picture:

- Flip horizontally about Y-axis
- Rotate left -30 degree
- Rotate right +30 degree

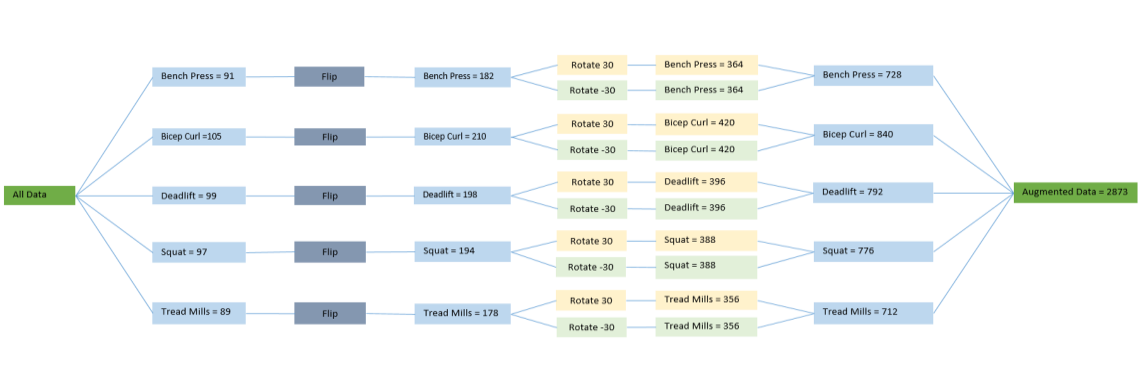

Fig 3.2: Data Augmentation

Fig 3.3: Flip Image

Fig 3.4: Rotate -30 and 30 degree rotate

3.5.2 Data Preparation

After augmentation for avoiding costs more computation resources and a chance of overfitting, we can reduce the input dimension fixed into 200 X 200 pixels of images. Before passing images to CNN we used RGB color that easily helps to detect features by CNN that ensure that to get better accuracy. For normalizing we can reduce RGB values dividing by 255 and get the range of [−0.5, 0.5].

3.6 Proposed Methodology

Convolutional neural network (CNN) is a specific type of artificial neural network architecture for deep learning that uses perceptron, a grating machine learning unit algorithm, for supervised learning, to analyze different types of data. CNN operations generally work depends on inputs for extracting pattern recognition and it works well with data that has a spatial relationship CNN also has a learnable parameter like neural network i.e., weights, biases, etc. [5]. Some of these layers are convolutional, using a mathematical operation and model to pass on results to successive layers.

3.6.1 Proposed Methodology Convolutional Layer

Convolutional neural network (CNN) is a specific type of artificial neural network architecture for deep learning that uses perceptron, a grating machine learning unit algorithm, for supervised learning, to analyze different types of data. CNN operations generally work depends on inputs for extracting pattern recognition and it works well with data that has a spatial relationship CNN also has a learnable parameter like neural network i.e., weights, biases, etc. [5]. Some of these layers are convolutional, using a mathematical operation and model to pass on results to successive layers.

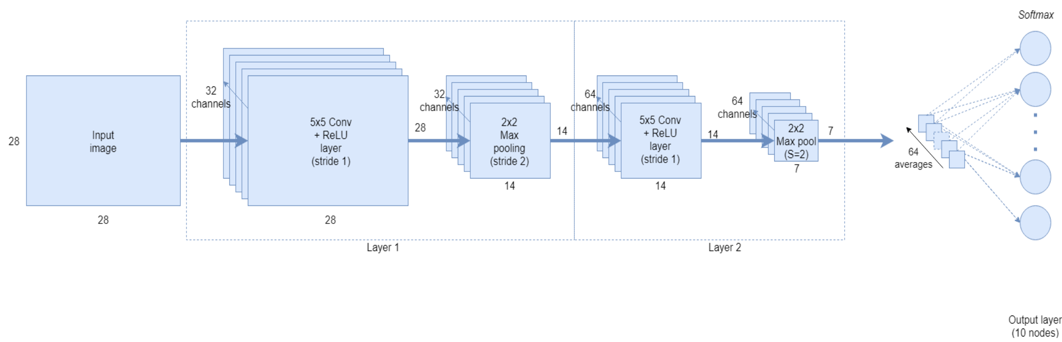

Fig 3.5: Convolutional neural network architecture

3.6.2 Convolutional layer

A convolutional neural network has a basic structure of an input layer, an output layer, and various hidden layers. Each input is convoluted with various types of the filter during the forward propagation (or kernel) [8]. We can apply in Image data, Classification prediction problems, Regression prediction problems, face recognition, object detection, segmentation, etc. There are several layers to classify an image such as CNN's first layer is the input layer that read-only image pixel to pixel, that image may be RGB of grayscale category. Then the second step uses filter to extract the feature of an input image in various ways. Different types of *pooling* layers use to reduce input image shape to reduce parameters. The result of the convolution shows momentum which affects the classification [8]. These samples are called features. To build a convolutional layer, some model hyper-parameters have to be configured: length of filters, number of filters, stride, and padding.

- Length of filters: Kernel acts as a filter, it's working for extracting specific features or patterns identifications in the input data, which make increase the efficiency of the classifications. In CNN's, filters are not defined. The value of every filter is learned throughout the training process. By humans able to learn the values of different filters, CNNs can find more meaning from input images by filtering but human-designed filters provably not be able to find specific features.

  A filter could be many different sizes like as 3X3 filter, which is small but probably easy to read input image pixels.

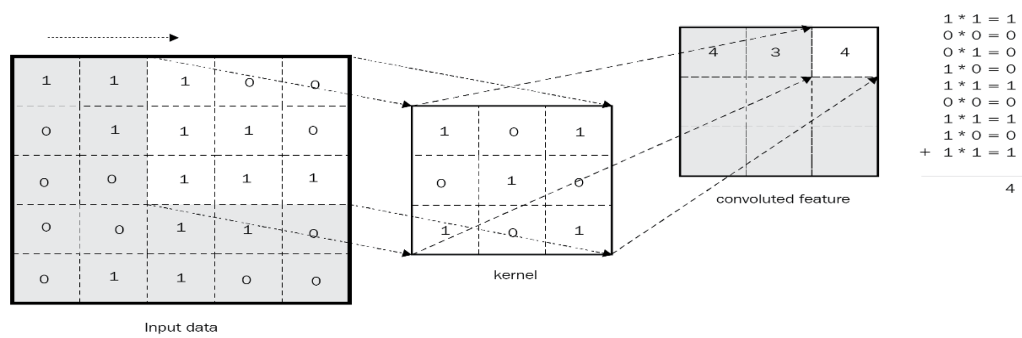

Fig 3.6: Kernel

- Stride: Stride defines the number of rows and columns that shifts pixels over the input matrix. Stride reduces the output dimension if the input matrix. If stride is 2 then move the filters to 2 pixels of the input matrix. Stride number always an integer and not a fraction, by default stride is 1.

- Padding: After the convolution layer reduces the dimensions of the output matrix, but using padding we can maintain the dimension of output as an input matrix. There are two kinds of padding: the same padding and valid padding. Valid padding means "no-padding", it reduces the dimension of the output matrix. The same padding means the output matrix is the same dimension as the input matrix. In the same padding adding an extra block and assign zero to the input matrix symmetrically for the same dimension.

Use the activation function (denoted σ) to identify those features that are relevant for classification after each convolutional operation [1].

3.6.3 Rectified Linear Units (ReLU)

Rectified Linear Units (ReLU): ReLU is the most widely used activation function while designing networks today. ReLU function is nonlinear and allows for backpropagation. The constant gradient of ReLUs results in faster learning because it does not activate all the neurons at the same time like as if the input is negative it will convert into zero and that neuron does not get activated. So few neurons are active at a time, not all neurons, at this reason ReLU much easier, faster and make more efficient. More biological inspired to train.

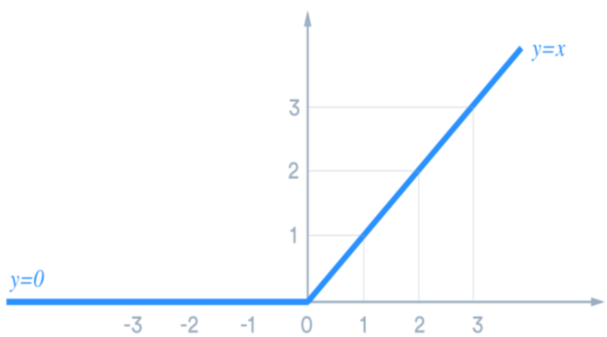

Fig 3.7: ReLU Activation Function

3.6.4 Pooling layer

The pooling layer is a non-linear layer that divides the dimension of the input and reduces the number of parameters, controlling overfitting and most relevant information is preserved. We can define the size of the PoolingLayer that can remove unnecessary features and keep the necessary features. There are three kinds of pooling layer *MaxPooling*, *AveragePooling*, and *MinPooling*. In deep learning, three pooling functions.

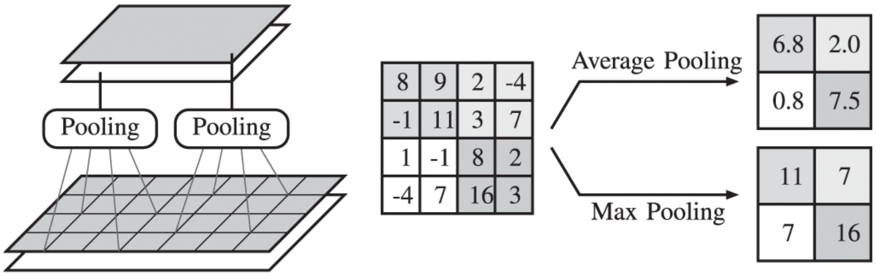

Fig 3.8: Pooling Layer

- MaxPooling: It picks only the maximum value contained in the pooling window. In CNN architecture MaxPooing is mostly used for recognizing relevant features because MaxPooling gives a better result from AveragePooling and MinPooling layer.
- AveragePooling: It picks only Average value contained in the pooling window.
- MinPooling: It picks only the Minimum value contained in the pooling window.

3.6.5 Flatten Layer

Flatten Layer In between the convolutional layer and fully connected layer, there's a 'Flatten' layer. It's converting all the resultant 2-dimensional arrays into a 1D feature vector, this operation is called flattening. This flattening structure makes a single long continuous linear vector to be used by the dense layer for the final classification layer.

3.6.6 Fully connected layer

A fully connected layer is the last phase for a CNN network, it represents the feature vector for the input. In FC layer involves weights, biases, and neurons. It connects neurons in one layer to neurons in another layer. FC layers recombine each neuron to efficiently and accurately classify each input. It is used to classify images between different categories by training. FC layers can be contrasted with Multilayer Perceptron (MLP) where each neuron has complete links with all previous layer activations [8].

3.6.7 Dropout layer

The dropout layer forces a neural network to learn more strong features that are useful in conjunction with many different random subsets of the other neurons [9]. This layer layers improve over-fitting, reduce dependence and complexity on the training set.

3.6.8 Softmax layer

The softmax layer is the last layer or output layer in neural network functions and its use for determining the probability of multiple classes. This function calculates the probabilities of each target class and returns the values to determine the target class for the given inputs [10].

3.7 Test Set

This dataset includes 5 different classes and contains an average of 95 core images for each class. After augmentation total of 2873 images for all classes. In this part, we make a test set to evaluate the classification performances of our CNN model. After data preprocessing we have split into two different portion test-set and train-set. First, train-set uses to train our model. When we successfully trained our model then we can evaluate our model using test-set. We select test-set and train-set using random state=42 to get more valid accuracy and performed well on the unseen test set. 75% of the 2873 images are used to create the training set and the remaining 25% is used for the test set [12]. So in train-set are contains a total of 2154 images and test-set 719 images.

3.8 Training the Model

After generating data preprocessing and defining train set, test set then we ready to train our model with training data sets that consist of 2154 images. For increasing accuracy and decreasing loss as possible, we update our model many times and change the optimizer, learning rate, loss function. We train our model using Adam optimizer to reduce loss function as possible and applied on a 75% training set and 25% validation set. We use 128 batch size for less memory and faster for training our model. As the process continues, then we can see that in 70 to 80 number of epochs training accuracy and validation accuracy are not increased significantly. At that stage validation accuracy reached 95.27%, training accuracy .9681%, and validation loss 0.1422%. Then our model is ready to predict unseen data for final test evolution.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 200, 200, 32)      2432
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 100, 100, 32)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 100, 100, 64)      18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 50, 50, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 50, 50, 96)        55392
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 25, 25, 96)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 25, 25, 96)        83040
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 12, 12, 96)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 13824)             0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               7078400
_________________________________________________________________
activation_1 (Activation)    (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 2565
=================================================================
Total params: 7,240,325
Trainable params: 7,240,325
Non-trainable params: 0
_________________________________________________________________
```

Fig 3.9: Model Summary

3.9 Implementation Requirements

After fully describing our CNN methodology and completely trained our mode a requisite list was created and those requirements are must essentially need for image-based classifications.

- Operating System (Windows 7 or above)
- Hard Disk (minimum 500 GB)

- Ram (Minimum 4 GB)
- GPU(Recommended)

Developing Tools

- Python Environment
- jupyter notebook (Anaconda3, Mini-Conda3)

CHAPTER 4

Experimental Results and Discussion

4.1 Introduction

In chapter 4, we discuss the performance evaluation of our model, the number of parameters, accuracy level. We compare our model with InceptionV3, VGG16, and MobileNet. For easily undergrad, we use graphs, pictures and confusion matrix.

4.2.1 Number of Parameters

In our convolutional network model decorated with some of the different layers, input size, number of filters, activation shape, etc. that generate weights, biases and that makes total 7,240,325 numbers of the parameter. The first input image shape is 200X200 using RGB color to read-only image without generating parameters. There are used 4 numbers of Conv2D layers and MaxPool layers one Flatten layer, 2 dense layers with one output layer or softmax layer. The pool size(2,2), strides(2,2) are same but number of filters(32,64,96,96) and kernel size (5,5), (3,3) are different in different layers.

Number 1: In the input layer used for reading image pixel to pixel. For Input_Size that's activation shape is (200X200X3) for that activation size are 120,000 because of input image wide 200, hight 200, and 3 for RGB color. In the input, the layer is no parameters.

Number 2: Conv2D 1 is the first layer of the convolutional network. It mainly works to extract features from an image with filters. In this model 32 number filter used and for that activation shape is (200X200X32) and activation size is 1,280,000. In Conv2D 1 generate (((5X5X3)+1)X32)=2,432 number parameters. Here (5X5) is kernel size and for RGB color 3.

Number 3: MaxPooling layer use to reduce the number of parameters, avoid the chance of overfitting. Here pooling size f=(2X2), striped s =2, Image wide w = 200, hight h = 200. So activation shape formula are (w-f+1)/s. so activation shape is ((200-2+1)/2) =99.5 =100. Pooling layer educe the image dimension 200 to 100 for that activation size is

(100X100X32)= 320,000. Here 32 is filter size. The pooling layer doesn’t generate parameters.

Number 4: Conv2D 2 is the second layer of our model. It complete the same work for activation size (100X100X64) =640,000 activation shape, and with 64 is number of filters make (((3X3X32)+1)X64)=2,432 18,496 number of parameters. Here (3X3) are kernel size, and 64 is filter size.

Number 4: In section 4 contain MaxPool 2. It is the second layer of MaxPool 2 for this model. At that same way, pooling layer reduces the number of image shape 100 to 50 and make activation size 160,000 without generating parameters.

Number 10: After the convolutional layer and pooling layer are flattened layer. It's converting all the resultant 2-dimensional arrays into a 1D feature vector. Activation shape (13824X1) = 13,824 number of activation size.

Number 11: In this section is a dense layer or fully connected layer. In dace, the layer has activation shape (512) but no activation size. It generates only parameters ((1+13824)X512) = 7078400.

Number 12: Softmax layer is the last layer of this model. We can say that it is the output layer of our model. The softmax layer generates parameters with a final shape of output. Here (5X(512+1)) = 2565 number of parameters.

After adding the total number of parameters is 7,240,325. CNN's Conv2D and MaxPool layers generate activation size using activation shape. MaxPool layer uses for reducing dimension size, and there are no parameters. Total parameters generate only Conv2D layers. For stable generalization, we don't increase CNN layers.

Table 4.1: Number of parameters

Layer		Number of Filters	Activation Shape	Activation Size	Parameters
Number	Operation				
1	Input Size	-	200,200,3	120,000	-
2	Conv2D 1	32	200,200,32	1,280,000	2,432
3	MaxPool 1	-	100,100,32	320,000	-
4	Conv2D 2	64	100,100,64	640,000	18,496
5	MaxPool 2	-	50,50,64	160,000	-
6	Conv2D 3	96	50,50,96	240,000	55,392
7	MaxPool3	-	25,25,96	60,000	-
8	Conv2D 4	96	25,25,96	60,000	83,040
9	MaxPool4	-	12,12,96	13,824	-
10	Flatten	-	13824,1	13,824	-
11	Dense 1	-	512	-	7078400
12	Softmax	-	5	-	2565
Total Parameters					7,240,325

4.2.2 Performance Evaluation

In order to find the best practices, we applied the proposed CNN architecture to the datasets described above and achieves significantly better results on average 95 per-class accuracies. In this picture, we can see the loss is going downwards, and at this time validation accuracy are increasing, so at that stage, we can say that the training model learns perfectly from the training data set.

Table 4.2: Classification Result

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Number of epochs
Inception V3	74	77.2	75.8	77	80
VGG16	87	87	87	86.6	80
MobileNet	90	92.2	89.4	90	80
CNN	95	96	95	95	80

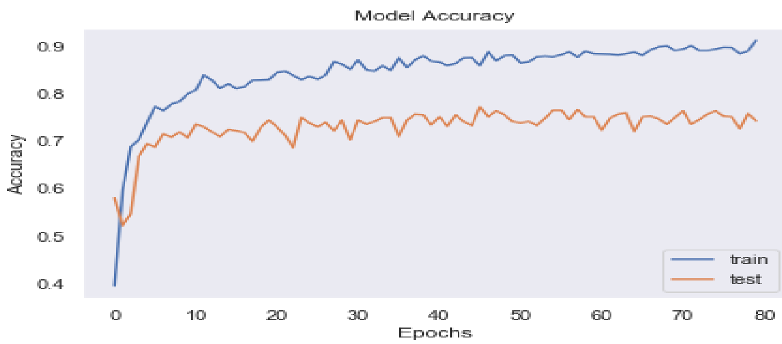

Fig 4.1: CNN Training and validation Accuracy

Fig 4.2: CNN Training and validation Loss

4.3 Result Discussion

We know that the classifier's performance was established on a test set from the training, validation and testing accuracy [13]. Our CNN model gives high accuracy of precision, recall and every weighted average up to 93. Total test dataset images are 719, and after classification, only 34 images are false predictions, another way 685 is a correct prediction. The final test of our model gives 95% accuracy, so our model gives a better test accuracy for unseen data. To make a clear assume we can observer the confusion matrix Fig 4.3.

Table 4.3: CNN Classification Report

Classes	Precision (%)	Recall (%)	F1 (%)
Bench Press	98	87	92
Bicep Curl	94	97	95
Deadlift	88	99	93
Squat	99	94	97
Tread Mills	98	100	99

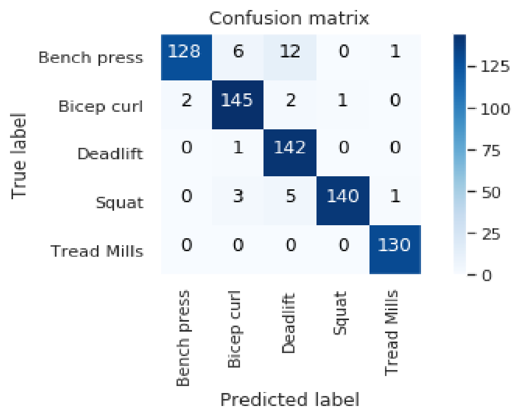

Fig 4.3: Confusion matrix

4.4 Comparison

In this section, we compared different models with our CNN models such as Inception-v3, VGG16, and MobileNet with their test accuracies and used the same number of epochs and batch-size are used for best comparison. For every model to train used 25% of test-set and 74% train-set are the same but validation accuracy and testing accuracy are different by a different model that is summarized in table 2[14]. From the table, we observed that VGG16 gives 87% accuracy with a little bit noisy, and validation accuracy and train accuracy rate often similar. MobileNet accuracy is 90% but validation accuracy and train accuracy are very noisy. InceptionV3 gives low validation accuracy for these datasets. But our CNN model gives the highest validation accuracy for a good train accuracy with a little bit noisy.

Finally, we observed that under the same conditions, every model's accuracy is satisfied, but among them, our CNN performs perfectly with a validation accuracy of 95%.

We proposed a new CNN architecture and that achieved the high testing accuracy from another model.

4.4.1 InceptionV3

Inception3 gives a good validation accuracy of 74% for our dataset. But in Fig 4.4.1.1 we can observe that training accuracy is good and increasing training accuracy and validation accuracy slowly. slowly but according to that validation, accuracy is not satisfiable. In 4.4.1.2 train loss is degreasing softly.

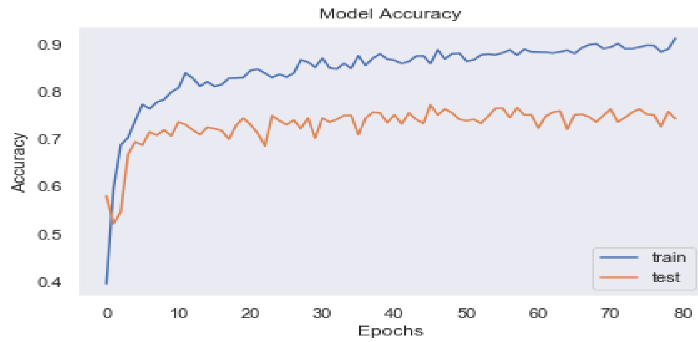

Fig 4.4: InceptionV3 Training and validation accuracy

Fig 4.5: InceptionV3 Training and validation Loss

4.4.2 VGG16

VGG16 performed well for our dataset and give 87% validation accuracy. In Fig: 4.4.2.1 we see that a little difference between training accuracy and validation accuracy. In training time it makes low noise and smoother. Fig: 4.4.2.2 shows the training loss of VGG16 it going to the lower loss.

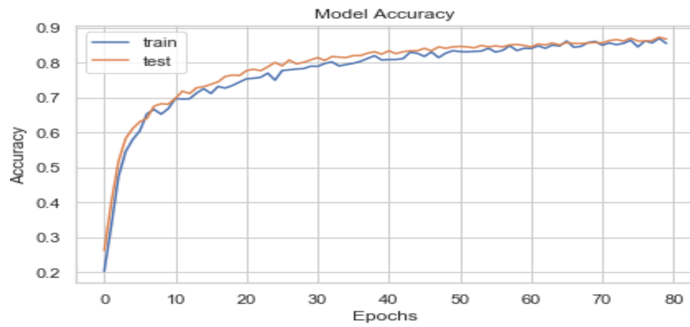

Fig 4.6: VGG16 Training and validation accuracy

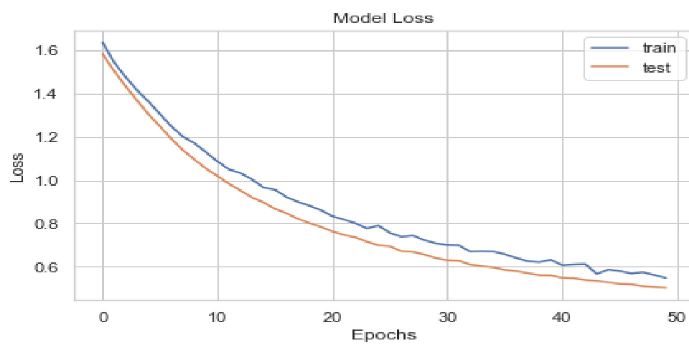

Fig 4.7: VGG16 Training and validation Loss

4.4.3 MobileNet

MobileNet gives 90% validation accuracy but it makes a very high noisy result for training accuracy and validation accuracy. In Fig 4.4.3.1 we show training accuracy is good but validation accuracy is very noisy. And in Fig 4.4.3.2 give the same result for training loss and validation loss.

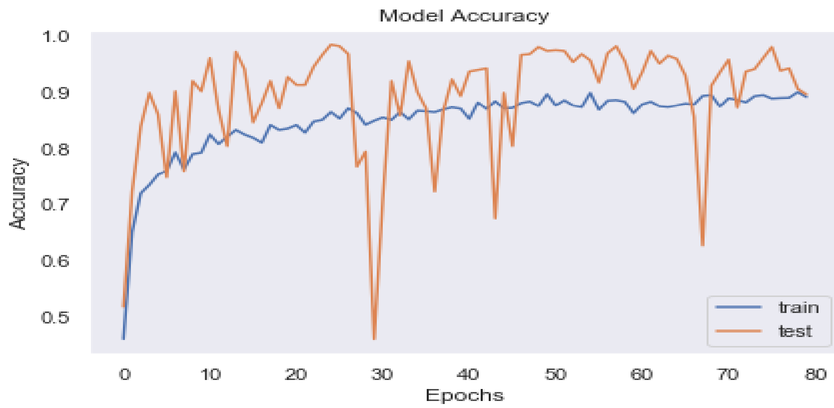

Fig 4.8: MobileNet Training and validation Accuracy

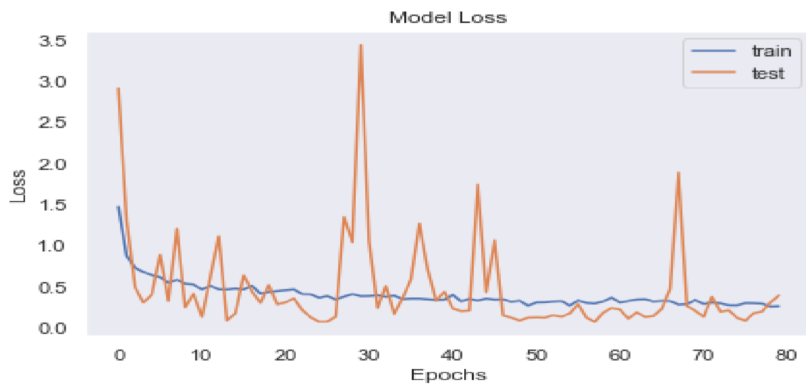

Fig 4.9: MobileNet Training and validation Loss

4.4.4 CNN

Our CNN model gives the best validation accuracy of 95% for our dataset. In Fig 4.4.4.1 we show that training accuracy and validation accuracy are satisfiable because it makes a little bit noise and successfully avoids overfitting. Fig: 4.4.4.2 shows that training loss is losing significantly. Our CNN gives high validation accuracy from InceptionV3, VGG16, and MobileNet.

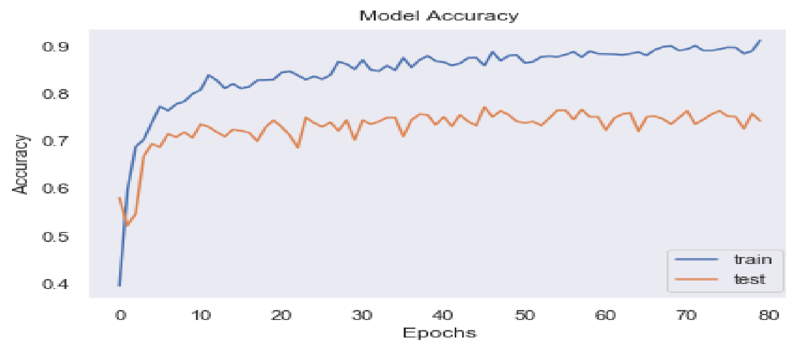

Fig 4.10: CNN Training and validation Accuracy

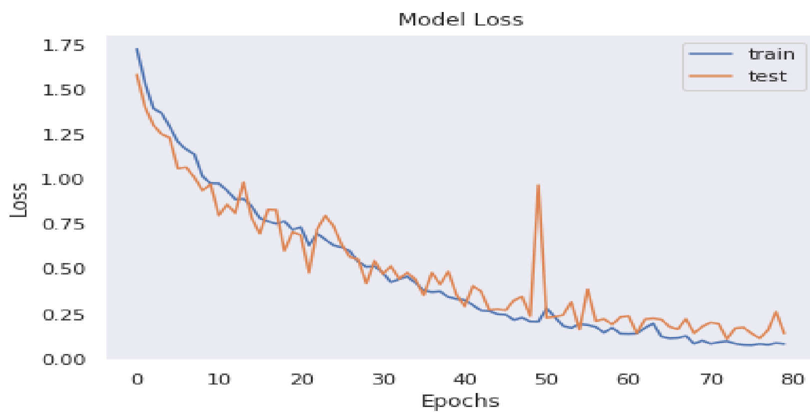

Fig 4.11: CNN Training and validation Loss

CHAPTER 5
Conclusion And Future Works

5.1 Conclusions

In this paper, we build up a model using CNN architecture and its competitive classification accuracy performance up to 95%. And we described the number of filters, Activation shape, Activation size, total parameters and number of convolutional blocks of our model. Our model is simple but performed much faster with high accuracy from another complex model. Finally, we demonstrate that the proposed method has high accuracy from the experiment. At last, we can say that CNN can be used to improve the object classification capacities of different areas in a new and innovative way [15].

5.2 Future Work

Our proposed model CNN shows better accuracy for classification against inceptionV3, VGG16, and MobileNet for different pose detection. But in the future, there are several ways to update our model for transfer learning. We'll apply another various models such as *AlexNet*, *ResNet* to increasing accuracy, efficient training and feature extraction for all models and other hand GPU are the most important for training. But we suggest a strong approach apply ensemble method for the best accuracy and we shall test other advanced classification ideas, such as transfer learning [11].

APPENDIX

We have faced so many problems to complete our project and that first was methodology selection. This project was so challenging because only some works done before with different datasets and different ways. We have to collect data from different online platforms, manipulate them to train our model. We made a new CNN model get a better classification accuracy rate and trained that successfully. We also trained some other traditional CNN models and compared them with our won model. Successfully our model gets the best classification result from other models. Our development task was very difficult but our thesis work was very interesting.

REFERENCES

[1] Haque, S., Rabby, A.S.A., Laboni, M.A., Neehal, N. and Hossain, S.A., 2018, December. ExNET: Deep Neural Network for Exercise Pose Detection. In International Conference on Recent Trends in Image Processing and Pattern Recognition (pp. 186-193). Springer, Singapore.

[2] Um, T.T., Babakeshizadeh, V. and Kulić, D., 2017, September. Exercise motion classification from large-scale wearable sensor data using convolutional neural networks. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 2385-2390). IEEE.

[3] Pishchulin, L., Andriluka, M., Gehler, P. and Schiele, B., 2013. Poselet conditioned pictorial structures. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 588-595).

[4] Toshev, A. and Szegedy, C., 2014. Deeppose: Human pose estimation via deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1653-1660).

[5] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C. and Torr, P.H., 2015. Conditional random fields as recurrent neural networks. In Proceedings of the IEEE international conference on computer vision (pp. 1529-1537).

[6] Chandra, S. and Kokkinos, I., 2016, October. Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs. In European Conference on Computer Vision (pp. 402-418). Springer, Cham.

[7] Long, X., Yin, B. and Aarts, R.M., 2009, September. Single-accelerometer-based daily physical activity classification. In 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (pp. 6107-6110). IEEE.

[8] Zaid, G., Bossuet, L., Habrard, A. and Venelli, A., Methodology for Efficient CNN Architectures in Profiling Attacks.

[9] Dropout in (Deep) Machine learning.<<https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5>> Last accessed on 11-09-2019 at 03.06 AM.

[10] Islam, M.S., Foysal, F.A., Neehal, N., Karim, E. and Hossain, S.A., 2018. InceptB: a CNN Based classification approach for recognizing traditional Bengali games. Procedia computer science, 143, pp.595-602.

[11] Zhang, Y.D., Dong, Z., Chen, X., Jia, W., Du, S., Muhammad, K. and Wang, S.H., 2019. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. Multimedia Tools and Applications, 78(3), pp.3613-3632.

[12] Kesim, E., Dokur, Z. and Olmez, T., 2019, April. X-Ray Chest Image Classification by A Small-Sized Convolutional Neural Network. In 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT) (pp. 1-5). IEEE.

[13] Maron, R.C., Weichenthal, M., Utikal, J.S., Hekler, A., Berking, C., Hauschild, A., Enk, A.H., Haferkamp, S., Klode, J., Schadendorf, D. and Jansen, P., 2019. Systematic outperformance of 112 dermatologists in multiclass skin cancer image classification by convolutional neural networks. European Journal of Cancer, 119, pp.57-65.

[14] Nagpal, C. and Dubey, S.R., 2019, July. A performance evaluation of convolutional neural networks for face anti spoofing. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[15] Vaibhav, K., Prasad, J. and Singh, B., 2019. Convolutional Neural Network for Classification for Indian Jewellery. Available at SSRN 3351805.