



HAND GESTURE RECOGNITION USING CONVOLUTIONAL AND RECURRENT NEURAL NETWORK



Prepared by
Ahmed Marof Attia
Alyaa Ahmed Seleem
Dina Hussien Essawy

Under supervision of
DR. Abdel moneam Isamil



Hand Gesture Recognition using Convolutional Neural Network

Ahmed Marouf*, Aliaa Ahmed Dina Hussein ***

* Faculty of graduate studies for statistical research, Cairo University

** Faculty of graduate studies for statistical research, Cairo University

*** Faculty of graduate studies for statistical research, Cairo University

[a7medm3rouf@gmail.com, alyaaahmed2@gmail.com, dina.alsayed@feps.edu.eg]

Abstract-Gesture Recognition is a major area in Human- Computer Interaction (HCI). HCI allows computers to capture and interpret human gestures as commands. A real-time Hand Gesture Recognition System is implemented and is used for operating electronic appliances. This research is an attempt to develop a gesture recognition system suitable for hand gestures using a smart TV. This system is implemented using the deep learning models such as the Convolution Neural Network (CNN) and the Recurrent Neural Network (RNN). The combined model will effectively recognize both static and dynamic hand gestures.

Keywords: Convolution Neural Network (CNN), Human- Computer Interaction (HCI), Recurrent Neural Network (RNN).

I. INTRODUCTION

The gesture is a means of communication through any physical movement which usually originates from the face or hand. No one knows the exact origin of gestures in human society when/how/why we started using gestures [2]. Gestures may be classified as dynamic or static. A dynamic gesture intends to vary over a period of time and static gesture tends to remain almost unchanged over time. This project emphasizes to recognize static gestures. However, the use of gestures as non-verbal communication is certainly limitless the goal of gesture recognition technology is to interpret human gestures through mathematical algorithms [1].

There are many techniques for hand gesture recognition such as the vision-based approach, glove-based approach, and depth-based approach. Gloves-based technologies require the user to wear devices-Sensors- that need a lot of cables to connect to the computer thus reducing ease of interaction, while Vision-based technologies take advantage of cameras and can capture characteristics such as hand texture and color to identify gestures [3].

As we said, there are many types of gestures such as arm gestures, faces, and many other hand gestures that give meaningful information. Hand gestures are a form of nonverbal communication that can be used in many areas such as communication between deaf and mute people, robot control, and medical applications [4]. The main purpose of the gesture recognition system is to develop a good system that can recognize human hand gestures and use them to control electronic devices [5]. and manage any household appliances in a simple and relatively inexpensive way such as lights, doors, fans, televisions, heaters, etc. [6]

The key problem is how to make a computer able to understand the hand gestures. Where Hand gestures vary in the direction of the fingers and the shape of the hands. the According to many researchers, it has been found that Convolutional Neural Network (ConvNet/CNN) is very effective when used to solve these problems [7][8] Therefore, this paper focuses on recognizing static hand gestures by building a model with a Conv3D & Conv2D+RNN that can analyze a large amount of image data and recognize static hand gestures.

In This paper We want to develop a cool feature in the smart-TV that can recognize five different gestures performed by the user which will help users control the TV without using a remote.



II. Related Work

Due to the importance of Hand Gesture systems and the applications based on it, many studies and researchers have been interested in activating models that categorize different gestures. The abundance of dataset contributed in increasing these papers as there are at least 50 hand gesture recognition datasets. They differ by the number of samples, their resolution, the number of classes, the presence of negative samples, the homogeneity of scenes, the distance between the camera. the following is an analysis of six studies divided between two trends: Application on Hand Gesture Recognition System and Hand Gesture Recognition using Convolutional Neutral Network.

II.1 Application on Hand Gesture Recognition System

Hand gesture recognition is one of the most advanced domains in computer vision and artificial intelligence, The application of the use of hand gestures can now be seen in games, virtual reality, and augmented reality, assisted living, cognitive development assessment etc. and also use to improve communication with deaf people but also to support gesture-based signaling systems. There are various studies on hand gesture recognition as the area is widely expanding, and there are multiple implementations involving both machine learning and deep learning.

One of these studies, Abdullah Mujahid and et.al. (1) proposes a lightweight model based on YOLO (You Only Look Once) v3 and DarkNet-53 convolutional neural networks for gesture recognition without additional preprocessing 'image filtering and enhancement of images. Data set consisted of 216 images. These images were further classified into 5 different Sets there were gestures like one, two, three, four, five. Each set held an average of 42 images, which were labeled using the YOLO labeling format. The proposed model achieved high accuracy even in a complex environment and it successfully detected gestures even in low-resolution picture mode. They used several different algorithms to test, which was the best method for the gesture recognition application. YOLO v3 was compared with other deep learning models VGG16 and SSD. In the end, YOLOv3 produced the best results and was chosen, with an accuracy of 97.68% during training and an outstanding 96.2% during testing.

In the same way, Rupesh Prajapati and et.al (2) use the Hand Gesture system to help deaf and dumb people, they made a project which aimed to develop a system that can convert the hand gestures into text. In other words, it focusses on placing the images in the database and with database, matching the image is converted into text. The detection involves observation of hand movement. The method gives output in text format that helps to reduce the communication gap between deaf-mute and people.

One of the most popular application in using Hand Gesture system is smart house system, Vinh Truong Hoang who use the HGM-4 dataset which contains total 4,160 color images (1280 × 700 pixels) of 26 hand gestures captured by four cameras at different position. It is organized in four main folders: CAM_Left, CAM_Right, CAM_Front and CAM_Below. There were 5 volunteers and each one performs 26 hand gestures and each camera caught 1040 mage. for the same implementation of smart homes, Phat Nguyen Huu and et.al, aimed to analyze and build an algorithm to recognize hand gestures applying to smart home applications. The proposed algorithm uses image processing techniques combining with artificial neural network (ANN) approaches to help users interact with computers by common gestures. They use five types of gestures, namely those for Stop, Forward, Backward, Turn Left, and Turn Right. Users will control devices through a camera connected to computers. The algorithm will analyze gestures and take actions to perform appropriate action according to users' requests via their gestures. The results show that the average accuracy of proposal algorithm is 92.6% for images and more than 91% for video, which both satisfy performance requirements for real-world application, specifically for smart home services. The processing time is approximately 0.098 second with 10 frames/sec datasets. However, accuracy rate still depends on the number of training images (video) and their resolution.

II.2 Why CNN

[1] Abdullah Mujahid and et.al," Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model", applied science, vol.11, No.4164, pp.1-15.

[2] Rupesh Prajapati and et.al., Hand Gesture Recognition and Voice Conversion for Deaf and Dumb" ", International Research (J Journal of Engineering and Technology , Vol. 5,No.4 , 2018,pp.1369-



According to various researchers, it has been found that ConvNets are very effective when they are used to solve problems related to computer vision. The biological similarity between the ConvNets and the human visual system is the main reason for the compatibility of ConvNet. The processing units of ConvNets consist of multiple layers of neurons that are distributed hierarchically. The parameters are shared among various neurons that are present in different levels in the network. This results in various connection patterns with the connections having varying weights. These weights are then used to complete the process of classification [6].

One of these studies, Savita Ahlawat and et.al that used CNN, which aimed to use the hand gesture application to recognize hand gestures with a webcam. The application using Convolutional Neural Network (ConvNet) for learning and classifying the hand gestures and was applied in five basic stages begin with Experimental Setup in which HD Webcam was used to scan the hand gestures of the user and to capture the images and using the Keras Library and Theano backend, second phase is Feature Extraction the interface which will capture the image and return the required features of the image is developed in Java. About the data collection 1000 images, 500 positional, and 500 rotational, are used to train the neural network for each hand gesture. Since there are 9 gestures (one being "none") there are 9000 images in the dataset. In The training phase multiple convolution layers and pooling layers which are used to compress the image and extract the important features which are used for training the neural network, ReLU activation is applied in each layer in convolutional layer and the accuracy of the recognition application is calculated from the observed values of matching percentage 90.125%.

In the same trend, Adithya V. and Rajesh R. used deep CNN () This paper proposes a methodology for the recognition of hand gestures, which is the prime component in sign language vocabulary to recognize sign language gestures which will be very beneficial to the deaf and dumb community, although there is a lot of studies using CNN to deal with hand gesture case, but The proposed model avoids the need for hand segmentation, which is a very difficult task in images with cluttered background also avoids the hectic job of deriving potential feature descriptor capable of recognizing the various gesture classes. about dataset, the National University of Singapore (NUS) hand posture data set with cluttered background containing 10 different hand posture classes with 200 images of each class. The postures are performed by 40 individuals of different ethnicities in complex natural backgrounds, the images are passed through a three-layered convolutional operation. The performance of the classification is evaluated with a fivefold cross validation and the confusion matrix is calculated with 94.7% accuracy, recall 94.85% and F-score 94.26%.

III. Proposed Model

A Gesture Recognition System is developed for controlling electronic appliances to provide remote access to electronic appliances without the use of switchboards. This Gesture Recognition system is implemented using four modules namely Dataset collection and Pre-processing, Training and Testing, Hand Detection, Feature extraction and Gesture Recognition.

The system uses a deep neural network which is the combination of Convolution Neural Network (CNN) and Recurrent Neural Network (RNN). Combining CNN and RNN will enhance the ability to recognize different actions at a varied time, so we can also recognize dynamic gestures.

The proposed system is built for application in smart home tv. The goal of this system is to build simple and common gesture data. The proposed gestures consist of five gestures, namely Increase Volume, Decrease Volume, jump forward 10 sec, Jump Backward 10 sec and pause. They are illustrated in flowing Fig:








Gesture ↕	Corresponding Action ↕
Thumbs Up	Increase the volume.
Thumbs Down	Decrease the volume.
Left Swipe	'Jump' backwards 10 seconds.
Right Swipe	'Jump' forward 10 seconds.
Stop	Pause the movie.

A. Data Collection and Pre-Processing

The dataset consisting of various hand gesture images at different conditions such as lightning, varied dimensions and so on are collected and are subjected to pre- processing. The pre-processing of the image consists of scaling, cropping, resize and normalization to eliminate redundant data, minimize data modification error.

Each gesture consists of a few hundred videos recorded by various people performing one of the five gestures in front of a webcam similar to what the smart TV will use. Each video recorded to 2-3 seconds long, but these videos entered into analysis as a sequence of 30 frames or images. So, this paper is considered as image classification problem.

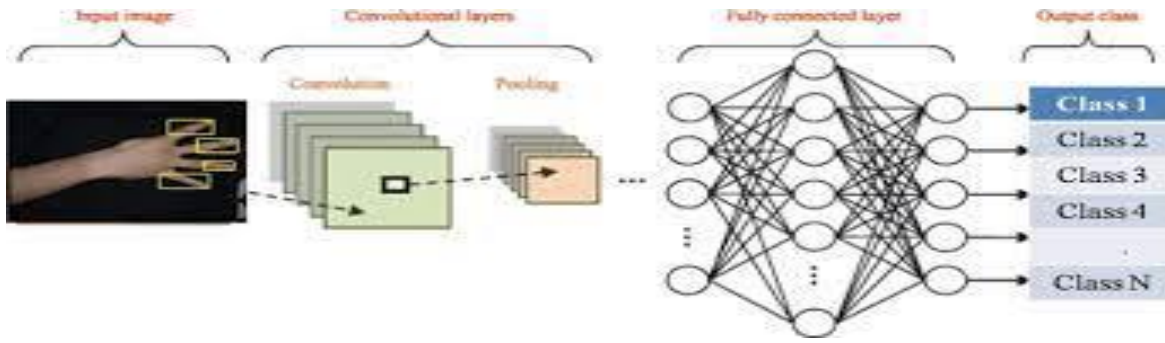
The Dataset divided between train and validation images, each folder of them consists videos- represented by images-for a specific gesture, the following table shows the number of images in each gesture:

Gesture	Image	Images No.in Training	Images No.in validation
Thumbs up		3870	450
Thumbs down (3)		4110	600
Left swipe (0)		4020	660
Right swipe (1)		4050	660
Stop (2)		3840	630

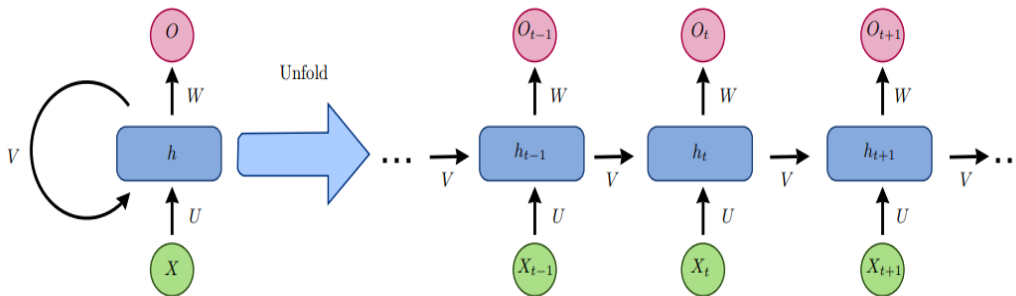
As the dimension of images are different either 360x360 or 160x120, Hence, we will need to do some pre-processing to standardize these images, so we create the generator, which has the task to do preprocessing steps like cropping, resizing and normalization.

Inside the Generator Function we create an empty 5D array as a batch data variable which will feed to our CNN and RNN model, then we iterate over the batch size (depending on our CPU or GPU) to fill the batch data variable using Python for and While loops.

CNN is the main method used in this paper, it is a type of Artificial Neural Network (ANN), which has deep feed-forward architecture and has amazing generalizing ability as compared to other networks with fully connected layers. Unlike other classical neural networks -where the input is in a vector format-in CNN the input is a multi-channeled image e.g. for RGB image ,so it is ideal for dealing with images and videos that learns relevant features from images, and performing several tasks like object classification, detection, and segmentation. [14] A convolutional layer is composed by a set of filters, also called kernels, that slides over the input data. Each kernel has a width, a height and width \times height weights utilized to extract features from the input data, Each filter in the convolutional layer represents a feature its task is trying to find a match to be activated, the CNN uses this process to learn the best filters to describe the objects. After that max pooling is applied in which the Kernel extracts the maximum value of the area it convolves. Max Pooling simply says to the Convolutional Neural Network that we will carry forward only that information, if that is the largest information available amplitude wise [15]. the following graph shows how CNN s working:



Recurrent Neural Network (RNN) is that the network contains at least one feed-back connection, so the activations can flow round in a loop. That enables the networks to do temporal processing and learn sequences, e.g., perform sequence recognition/reproduction or temporal association or prediction [16]. RNN unlike CNN and Feedforward nets, information travels from layer to layer with existing loops in the network so that each state is influenced by its previous states. Owing to the addition of loops, RNNs have a memory ability that allows them to store past computations and to exhibit dynamic temporal behavior RNNs have hidden layers with an additional self-looped connection that allows hidden layers at one time instant to be used as input for the hidden layer at the next time instant[17].that what is shown in the following figure:



B. Model

In CNN we start training on a small amount of data and then proceed further. So, we build 10 models by attempting to tuning its parameters, in first one we only use 16 Images in each 20 epochs with 64 batch size. Second model, we expand images to be 30 frames in 30 epoch with the same batch size, the same is used for third model 30 Image and 20 Epoch with reducing other parameter, change the epoch to 20 with the same parameter and frames of third model to do the fourth one. In the fifth model, we keep 20 frames in 20 Epochs, with reducing kernel to (2,2,2), switching Batch Normalization before Maxpooling. we also build four models use RNN, first one using Conv2D and LSTM, for another one we transfer Learning (Mobile Net) with LSTM.

CNN models that has been considered in this paper, is CNN 3D which applies a 3-dimensional filter to the dataset and the filter moves 3-direction (x, y, z) to calculate the low-level feature representations. Their output shape is a 3-dimensional volume space such as cube or cuboid. They are helpful in event detection in videos. They are not limited to 3D space but can also be applied to 2d space inputs such as images. The CNN that has been considered in this paper to has the same structure as a base model, which composed of three convolution layers, three max pooling layers, two fully connected layers and output layer.

We develop various models in CNN and RNN then calculate the accuracy and loss in training and validation, to decide which model is able to train without minimum errors and high accuracy.



We firstly tried to use 16 images as frames and 20 Epochs, another we tried 30 images and 30 Epochs, then 30 images in 20 Epochs.

The First convolution layer (input layer) has 16 different filters with the kernel size 3. In every convolution layer, ReLU activation is applied. The ReLU activation function is as follows:

$$Relu(x) = \text{maximum}(0, x)$$

That is, if $x < 0$, $Relu(x) = 0$ and if $x \geq 0$, $Relu(x) = x$.

This layer produces the feature maps and passes them to the next layer. Then the CNN has a max pooling layer with pool size 2 which takes the maximum value from a window of size 2. The spatial size of the representation is reduced progressively as the pooling layer takes only the maximum value and discards the rest. This layer helps the network to understand the images better because it only selects more important features.

The next layer is another convolution layer and it has 32 different filters with the kernel size 3, Again, ReLU was used as the activation function in this layer. This layer is followed by another max pooling layer which has a pooling size In this layer, third convolutional layer has the same kernel size (2), activation function (ReLU), maxpooling (2), stride (2) but filter size is 64.

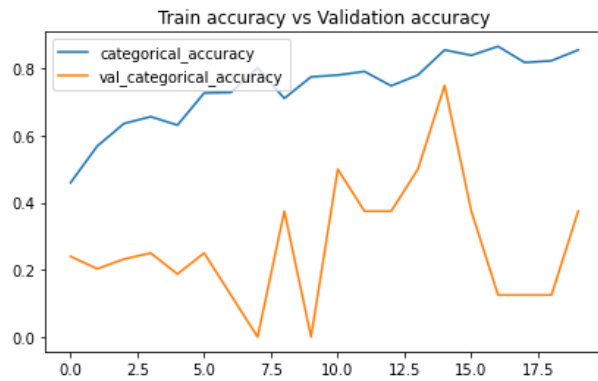
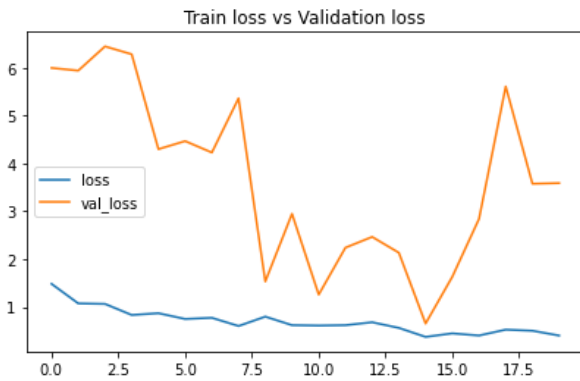
Output from the previous layers are received by the flattening layer, the next layer is fully connected layer which has 128 nodes and ReLU was used as the activation function and drop out is 0.25. and the second fully connected layer has 64 nodes and the same activation function and drop out. The output layer has 5 nodes corresponding to each classes of the hand gestures with using SoftMax function.

We have used Adam optimizer with its default settings. also use ReduceLROnPlateau callback enables to tweak the learning rate in the middle of the model fit and helps the model in faster convergence and leading the model to the global minima faster by reduce our learning alpha after 2 epochs on the result plateauing. Also, we used the concept of Transfer learning and use a pre-trained model and then use a GRU and LSTM layer for final training. Among the models available in the Keras API (Keras Applications), Mobile Net, DenseNet-169 were chosen because they are relatively low in parameters compared to other models, and the architecture handles the vanish gradient problem well

B.1: Model 1 &2:

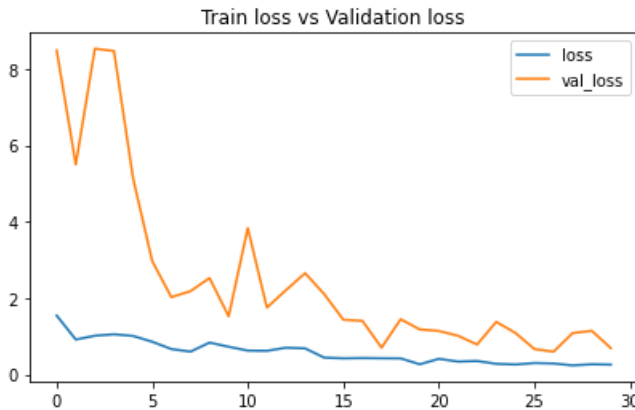
In those models we use 16 images in 20 Epochs, 30 images in 30 Epochs. The First convolution layer (input layer) has 16 different filters with the kernel size 5. The activation function ReLU. This layer produces the feature maps and passes them to the next layer then CNN maxpooling layer with pool size 2. The next layer is another convolution layer and it has 32 different filters with the kernel size 3, Again, ReLU was used as the activation function in this layer. This layer is followed by another max pooling layer which has a pooling size in this layer, third convolutional layer has the same kernel size (2), activation function (ReLU), maxpooling (2), stride (2) but filter size is 64. Output from the previous layers are received by the flattening layer, then feed to fully connected layer which has 128 nodes and ReLU was used as the activation function, followed with another second fully connected layer has 64 nodes and the same activation function and drop out. The output layer has 5 nodes corresponding to each classes of the hand gestures with using SoftMax function.

To assess this model, the accuracy and loss in training and validation data are calculated:



Model 1

The Training Accuracy was 0.866 with Validation Accuracy 0.75



Model 2

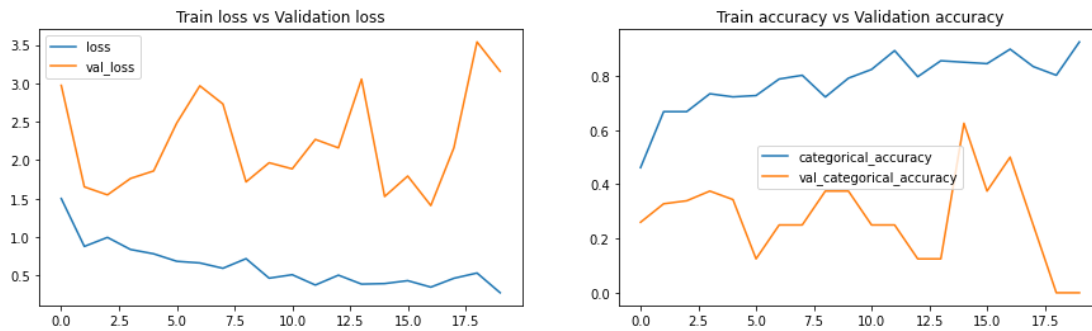
The Training Accuracy was 0.930 with Validation Accuracy 0.875

B.3: Model 3:

In this model we use 30 images in 20 Epochs, Reduced kernel, The First convolution layer (input layer) has 16 different filters with the kernel size 3. The activation function used in this layer is Rectified Linear Unit (ReLU). This layer produces the feature maps and passes them to the next layer. Then the CNN has a max pooling layer with pool size 2 which takes the maximum value from a window of size 2. The spatial size of the representation is reduced progressively as the pooling layer takes only the maximum value and discards the rest. This layer helps the network to understand the images better because it only selects more important features.

The next layer is another convolution layer and it has 32 different filters with the kernel size 3, Again, ReLU was used as the activation function in this layer. This layer is followed by another max pooling layer which has a pooling size in this layer, third convolutional layer has the same kernel size (2), activation function (ReLU), maxpooling (2), stride (2) but filter size is 64.

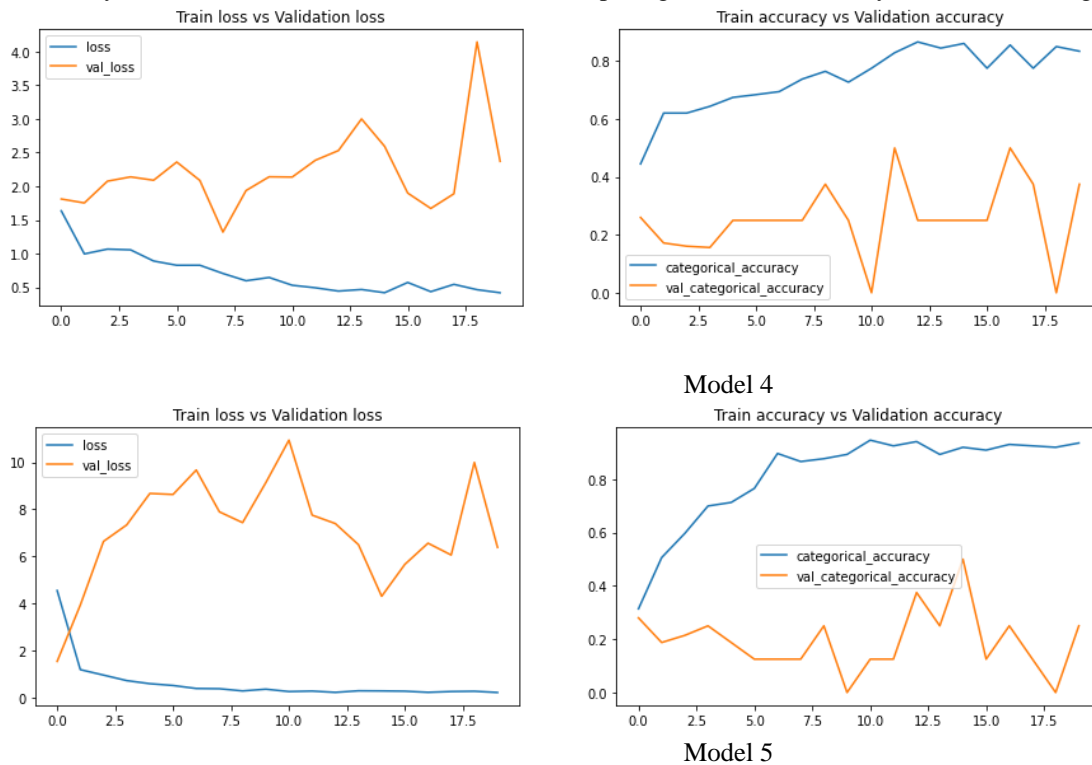
Output from the previous layers are received by the flattening layer, the next layer is fully connected layer which has 128 nodes and ReLU was used as the activation function and drop out is 0.25. and the second fully connected layer has 64 nodes and the same activation function and drop out. The output layer has 5 nodes corresponding to each classes of the hand gestures with using SoftMax function. To assess this model, the accuracy and loss in training and validation data are calculated:



These graphs showed that, the model perform good to some extent in training phase but not in that performance in validation data, that consistent with that matrixing Accuracy value is 0.93 and Max. Validation Accuracy 0.625, which indicate to the beginning of over fitting problem.

B4: Models 4&5:

Model four and five are too similar except the kernel size and switching batch normalization before Maxpooling happened in model five, we use four layers of Conv3D with sizes 16, 32, 64 and 128, respectively in model four and also in model five. ReLU activation function is four-layer MaxPooling3D in size 3 and kernel 3x3x3 but in another model's size 2. And one Flatten layer, respectively. The output is a dense layer of size 128 and another one in size 64 to output 5 gestures. the accuracy and loss in training and validation data are computed:

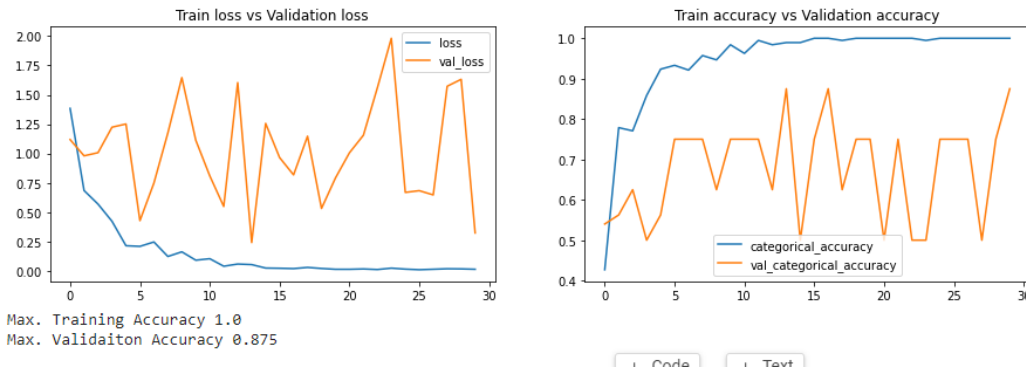


The previous figures show that the features of a over fitting problem has become more obvious, as the accuracy of model 4 and 5 is get more higher accuracy in training than in validation or in new data, in other word these model learn very well in training data but have a poor performance if their new data never been seen before from these model. So the Maximum Training Accuracy is 0.87 and Max. Validation Accuracy is only 0.5 for model 4, and reach to 0.95, 0.5 in model 5 respectively.



We used 20 frames, 30 Epochs, and batch size was 64, we used Time Distributed, create a Mobile Net "model" train with our dataset and classified, and did not train any layer, create a Sequential model, Then we add Flatten layer and 1 LSTM layers and drop out is 0.2

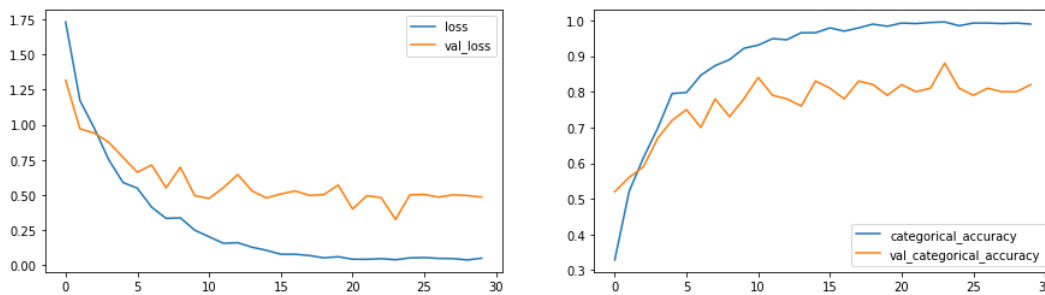
then add NN (Dense layer with a fully connected layer which has 256 nodes and ReLU was used as the activation and drop out is 0.2 and the output layer with SoftMax has num_classes neural, then we Compiled and Print the Summary of the Network Model and Create a Callback proved by Keras and Save the best Training Model Weights finally Analyzes the above model fit history and the best model checkpoint



B6: Model 8:

Here We added dense net layer then added Flatten layer, GRU Layer with 128 node and drop out is 0.25. Flatten layer and 1 LSTM layers and drop out is 0.2. then add NN (Dense layer with a fully connected layer which has 128 nodes and ReLU was used as the activation and drop out is 0.25 and the output layer with SoftMax has num_classes neural, then we Compiled and Print the Summary of the Network Model and Create a Callback proved by Keras and Save the best Training Model Weights finally Analyzes the above model fit history and the best model checkpoint

loss: 0.0399 - categorical_accuracy: 0.9955 - val_loss: 0.3262 - val_categorical_accuracy: 0.8800



IV. Results

Gesture Recognition System can be implemented using various techniques and some of them are listed under the literature survey. More than one using the Convolution Neural Network for recognizing the gesture just changing the epochs and frames numbers, and in another model, we use the RNN, the following table is briefing the results:

Experiment Number	Model	Result	Decision + Explanation
1	Conv 3D Model using 16frames per video + 16, 32, 64, 128 filters conv 3D layers + 128 dense nodes	Training loss: 0.4014 Training Accuracy: 86%	Mid loss and Mid to high accuracy Parameters - ~ 2,067,621.

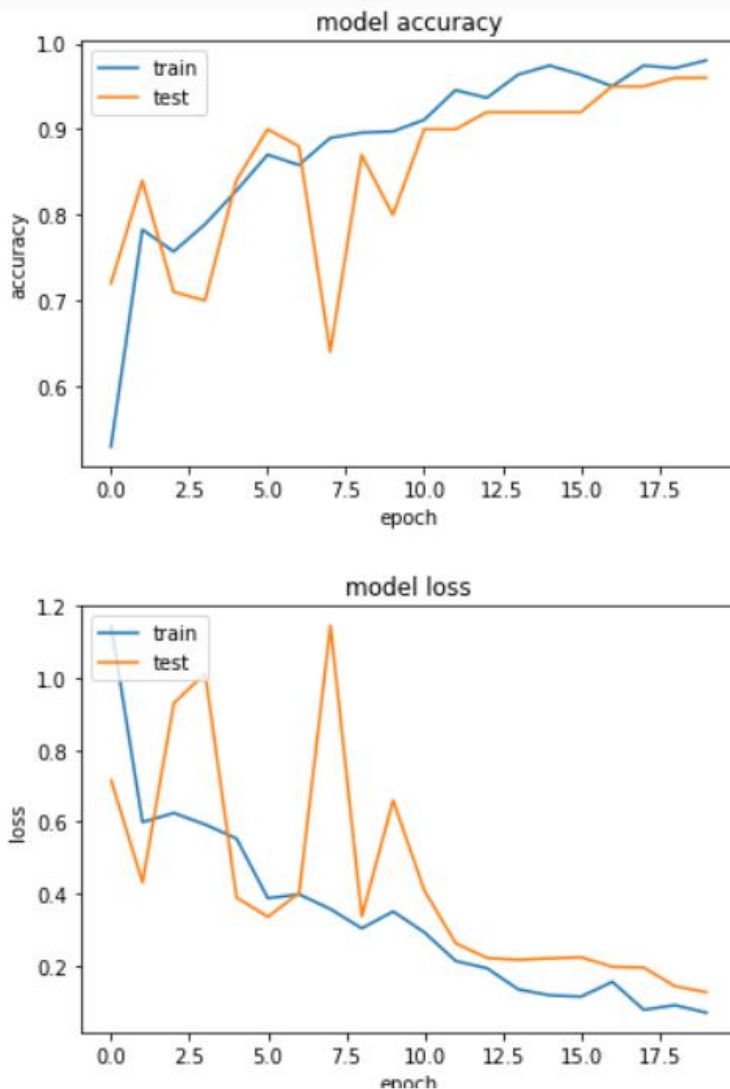


	+ 64 dense nodes + image size 120 by 120	Validation Accuracy: 75%	
2	Conv 3D Model using 20 frames per video + 16, 32, 64, 128 filters conv 3D layers + 256 dense nodes + 128 dense nodes + image size 120 by 120	Training loss: 0.2800 Training Accuracy: 93% Validation Accuracy: 87%	Low loss and high accuracy and comparable between training and validation data sets. Parameters - ~ 9,006,245.
3	Conv 3D Model with 30 frames per video + 16, 32, 64, 128 filters conv 3D layers + 256 dense nodes + 128 dense nodes + Batch_size=64	Training Accuracy: 93% Validation Accuracy: 62.5%	Parameters are on higher side and even the model accuracy is not good enough. Parameters~~ 5,617,637 million Model over fitting.
4	Conv 3D Model with 20 frames per video + 16, 32, 64, 128 filters conv 3D layers + 256 dense nodes + 128 dense nodes + Same pooling	Training Accuracy: 87% Validation Accuracy 50%	Parameters: 1,907,045 There is over fitting problem
5	Conv 3D Model with 20 frames per video + 16, 32, 64, 128 filters conv 3D layers + 256 dense nodes + 128 dense nodes reduced kernel to (2,2,2) switching Batch Normalization before Max Pooling	Training Accuracy: 95% Validation Accuracy: 59%	Parameter 1,301,045 There is over fitting problem
6	Conv2D+LSTM Model with 20 frames per video + 16, 32, 64, 128, 256 filters conv 2D layers + LSTM (256) + Dense (256 nodes)	Training Accuracy: 93% Validation Accuracy: 68%	Parameter: 3,084,133 Model going over fitting.



	+ Epochs = 20		
7	Conv2D+LSTM Model with change hyper parameters 20 frames per video 32, 64, 128,256 ,256 filters conv 2D layers + add 2 LSTM (128) + Dense (128 nodes) + Epochs = 30	Training Accuracy: 90% Validation Accuracy: 74%	Parameter 1,324,869
8	(Mobile Net) with LSTM + LSTM (256 cells) + Dense (256 nodes) ++ 20 images per video	Training Accuracy: 1% Validation Accuracy: 87% Training loss: 0.0137	Total params: 4,611,781 Trainable params: 1,380,869 Non-trainable params: 3,230,912 Low loss and high accuracy
9	DenseNet169 + GRU (128 cells) + Dense (128 nodes) + 20 images per video	Training Accuracy: 99% Validation Accuracy: 88% Training loss: loss: 0.0399	Total params: 18,460,741 Trainable params: 5,858,053 Non-trainable params: 12,602,688 Low loss and high accuracy
10	Mobile net with LSTM + LSTM (128 cells) + Dense (64 nodes) + 20 images per video + random data transformations on the images	Training Accuracy: 98% Validation Accuracy: 93% Training loss: 0.0696	Total params: 3,411,871 Trainable params: 1,031,002 Non-trainable params: 2,380,869 Low loss and high accuracy

And for the best model accuracy it was Mobile net + LSTM, and it's plot as below:



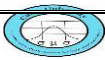
V. Conclusion

The power of deep learning tool for the recognition of hand postures from raw images (RGB) has been utilized in this work. In this research, Gesture Recognition dataset was used for recognizing five different gestures performed by the user which will help users' control. The proposed of 3D Conv model is study of ability to predict the 5 gestures correctly. Besides, the CNN-LSTM model for gesture datasets has been compared with two pre-trained models namely, Mobile Net, DenseNet169. To avoid overfitting, batch normalization and L2 regularization has been used.

It has been demonstrated in the result section of the Mobile net with LSTM model outperforms other models in terms of accuracy. It achieved 98% training, 93% validation. The proposed model in the future, more advanced deep learning techniques will be applied for human gesture recognition.

VI. Future Work

There is research that aims to build a dynamic data set by engaging more categories of gestures and collecting images from a larger number of subjects. Si-Jung Ryu's paper [7] proposes a hand gesture recognition system for a real-time implementation of HCI using 60 GHz continuous wave (FMCW) radar, developed by Google. The overall system includes a signal processing part that generates clutter-free Range Doppler Map (RDM) sequences and a machine learning part including a long-term memory (LSTM) encoder to learn the temporal properties of RDM sequences. A set of data is collected from 10 experimental participants. The proposed hand gesture recognition system successfully distinguishes 10 gestures with a high classification accuracy of 99.10%. It also recognizes the new subscriber's gestures with an accuracy of 98.48%.



Also, in recent years, recent improvements in imaging sensors and computing units have led to the development of a range of human-machine interaction interfaces (HMIs). An important approach in this direction is the use of dynamic hand gestures of the gesture-based interface

To this end, Seunghyeok Shin [8] used PB-GRU-RNN for skeleton-based dynamic hand gesture recognition after noise removal, data normalization, feature fragment segmentation, and feature extraction. As a result, they got better discrimination performance than most existing methods. In contrast to current methods that used the entire feature for input, their method divided the features into multiple parts and used them as input to GRU-RNNs for each of the parts of the hand.

This reduced the number of parameters required for neural networks and improved performance; Therefore, less memory is required to create HMI systems with neural networks.

REFERENCES.

- 1) C. Chansri, J. Srinonchat, "Hand Gesture Recognition for Thai Sign Language in Complex Background Using Fusion of Depth and Color Video", *Procedia Computer Science* 86 (2016) 257–260, doi: 10.1016/j.procs.2016.05.113.
- 2) D.-L. Dinh, J.T. Kim, T.-S. Kim, "Hand Gesture Recognition and Interface via a Depth Imaging Sensor for Smart Home Appliances", *Energy Procedia* 62 (2014) 576–582, doi: 10.1016/j.egypro.2014.12.419.
- 3) F. Dominio, M. Donadeo, P. Zanuttigh, "Combining multiple depth-based descriptors for hand gesture recognition", *Pattern Recognition Letters* 50 (2014) 101–111, doi: 10.1016/j.patrec.2013.10.010.
- 4) Haiying Guan, Jae Sik Chang, Longbin Chen, R.S. Feris, M. Turk, Multi-view Appearance-based 3D Hand Pose Estimation, in: *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, New York, NY, USA, IEEE, 2006 154–154, doi: 10.1109/CVPRW.2006.137
- 5) A. Just, S. Marcel, "A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition", *Computer Vision, and Image Understanding* 113 (2009) 532–543, doi: 10.1016/j.cviu.2008.12.001.
- 6) A.I. Maqueda, C.R. del-Blanco, F. Jaureguizar, N. García, "Human-computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns", *Computer Vision and Image Understanding* 141 (2015) 126–137, doi: 10.1016/j.cviu.2015.07.009.
- 7) Z. Ren, J. Yuan, Z. Zhang, Robust Hand Gesture Recognition Based on Finger-Earth Mover's Distance with a Commodity Depth Camera, in: *Proceedings of the 19th ACM International Conference on Multimedia*, Association for Computing Machinery, New York, NY, USA, 2011, pp. 1093–1096, doi: 10.1145/2072298.2071946.
- 8) P.K. Pisharady, M. Saerbeck, Recent methods and databases in vision-based hand gesture recognition: A review, *Computer Vision, and Image Understanding* 141 (2015) 152–165, doi: 10.1016/j.cviu.2015.08.004.
- 9) G. Poon, K.C. Kwan, W.-M. Pang, Real-time Multi-view Bimanual Gesture Recognition, in: *2018 IEEE 3rd International Conference*.
- 10) P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3d convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 1–7.
- 11) C. J. L. Flores, A. G. Cutipa, and R. L. Enciso, "Application of convolutional neural networks for static hand gestures recognition under different invariant features," in *2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, IEEE, 2017, pp. 1–4.
- 12) M. W. Cohen, N. B. Zikri, and A. Velkovich, "Recognition of continuous sign language alphabet using leap motion controller," in *Proceedings of the 2018 11th International Conference on Human System Interaction (HSI)*, pp. 193–199, Gdansk, Poland, July 2018
- 13) T.-N. Nguyen, H.-H. Huynh, and J. Meunier, "Static hand gesture recognition using principal component analysis combined with artificial neural network," *Journal of Automation and Control Engineering*, vol. 3, no. 1, 2015.
- 14) Anirudha Ghosh, Abu Sufian and Farhana Sultana, *Fundamental Concepts of Convolutional Neural Network*, in book: *Recent Trends and Advances in Artificial Intelligence and Internet of Things*, 2020, pp.519-567.
- [15] Jawad Nagi, "Max-pooling convolutional neural networks for vision-based hand gesture recognition", *Conference: 2011 IEEE International Conference on Signal and Image Processing Applications, ICSIPA 2011*, Kuala Lumpur, Malaysia, 2011 November 16-18.
- [16] Younes Ed-Doughmi, Najlae Idrissi and Youssef Hbali, "Real-Time System for Driver Fatigue Detection Based on a Recurrent Neuronal Network", *J. Imaging*, Vol.(6), No.(8), 2020, pp.1-14.
- [17] Robert DiPietro and Gregory D. Hager, "Chapter 21 - Deep learning: RNNs and LSTM" in *Handbook of Medical Image Computing and Computer Assisted Intervention*, 2020, pp.503-519.
- [18] Choi, Jae-Woo, Si-Jung Ryu, and Jong-Hwan Kim. "Short-range radar based real-time hand gesture recognition using LSTM encoder." *IEEE Access* 7 (2019): 33610-33618.
- [19] Shin, Seunghyeok, and Whoi-Yul Kim. "Skeleton-based dynamic hand gesture recognition using a part-based GRU-RNN for gesture-based interface." *Ieee Access* 8 (2020): 50236-50243.



AUTHORS

Author – Ahmed Marouf, *Petroleum Engineering bachelor's degree, Cairo University, FGSSR, a7medm3rouf@gmail.com.*

Author – Dina Hussein, *Cairo University, FGSSR, dina.alsayed@feps.edu.eg.*

Author – Aliaa Ahmed, *Cairo University, FGSSR,, alyaaahmed2@gmail.con.*