International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 08 | Aug 2021 www.irjet.net p-ISSN: 2395-0072

Identification of Biomedical Instruments Using CNN Architecture

Prashant Dhope[1]

[1]*PG Scholar, Dept. of E&C Engineering, Center for PG Studies, VTU-Belagavi, Karnataka, India*

Abstract – *The biomedical instruments plays a very important role in medical field that helps the physicians to diagnose and provide treatment for the problem. In today's era, there are lots of advancements happening in the field of deep learning. This paper represents the identification of biomedical instruments which are used in medical fields using the convolutional neural network (CNN) algorithm. The study consists of self prepared dataset of twelve different biomedical instruments for training and validation purpose. From the experimental results, the proposed methodology provides the training and validation accuracy of 79.56% and 57.42% respectively.*

Key Words: Biomedical Instruments, Biomedical Engineering, Convolutional Neural Network, Deep Learning, Self Prepared Dataset

1. INTRODUCTION

Biomedical Instrumentation is an application of Biomedical Engineering that focuses on the instruments and mechanics utilized to compute, assess and serve the biological systems. Various biomedical instruments are utilized in the medical sector for treatment or diagnosis purpose. The biomedical instruments make use of multiple sensors to monitor the physiological parameters of a human being.

Nowadays with the aid of Artificial Intelligence (AI), machine and human could able to communicate with each other. The branches such as Machine Learning (ML) and Deep Learning (DL) are contributing a lot for making the intelligent healthcare or medical sectors. The best example is the machines like robots could able to perform the operations or surgery without the assistance of doctor or physician.

The aim of this study is to build the model for identification of various biomedical instruments using the proposed convolutional neural network architecture. The rest of the paper is structured as follows: section-2 presents the proposed methodology for developing a model, section-3 highlights the experimental results and finally section-4 concludes the study.

2. METHODOLOGY

Fig -1: Block Diagram of Proposed Methodology

The block diagram of proposed methodology is depicted in Fig. 1 that consists of mainly five stages: (i) Dataset Creation, (ii) Data Pre-processing, (iii) Data Splitting, (iv) CNN Design and (v) Training & Validation phase.

2.1 Dataset Creation

In this study, the self prepared dataset is used. The dataset contains total 517 images of twelve biomedical instruments such as Audiometer, Cannula, CT scanner, Defibrillator, Dialyzer, Digital BP meter, Enema bulb, Needle electrode, Ophthalmoscope, Sphygmomanometer, Stethoscope and Syringe. All the images of dataset are downloaded using the Python's simple_image_download package. The Fig. 2 depicts the sample images from the dataset.

Fig -2: Sample Images from the Dataset

2.2 Data Pre-processing

In this stage, all the images from the dataset are converted into grayscale, which minimizes the computational requirements of the algorithm. Then all the images are resized to a fixed dimension of 48 x 48 (width x height).

2.3 Data Splitting

In this stage, the entire dataset is divided into two parts as Training and Validation. The 70% of the data is used for training and remaining 30% data is used for validation purpose. After train-validation split, the training dataset contains 362 images, whereas validation dataset contains 155 images.

2.4 CNN Design

The sequential model is used for building the CNN architecture. The proposed structure of CNN is depicted in Fig. 3, which consists of four convolutional layers with kernel size of 3 x 3, two max-pooling layers with pool size of 2 x 2, one flattening layer and two dense layers. Finally, the Softmax activation function is used that classifies the biomedical instruments. The Fig. 4 highlights the summary of proposed CNN architecture.

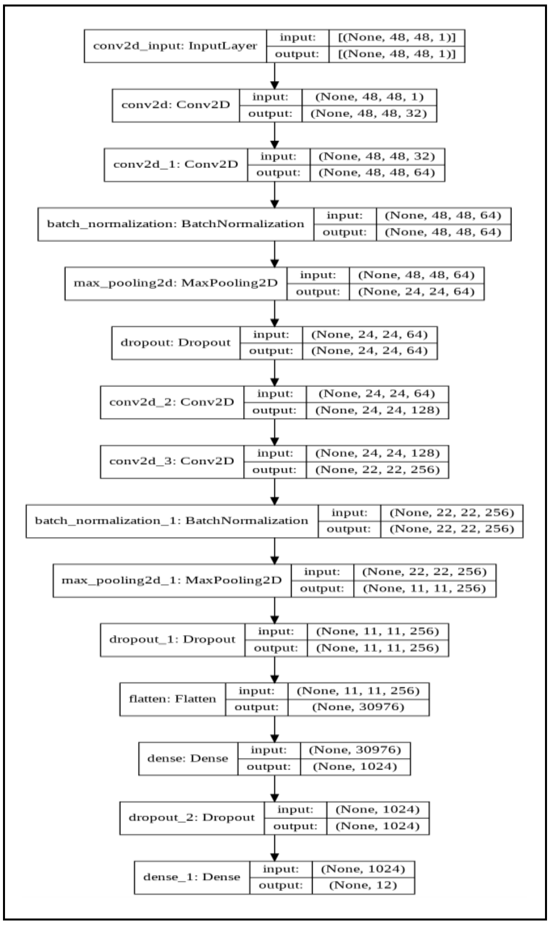

Fig -3: Proposed Structure of CNN

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 48, 48, 32)        320
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 48, 48, 64)        18496
_________________________________________________________________
batch_normalization (BatchNo (None, 48, 48, 64)        256
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 24, 24, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 24, 24, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 128)       73856
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 22, 22, 256)       295168
_________________________________________________________________
batch_normalization_1 (Batch (None, 22, 22, 256)       1024
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 11, 11, 256)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 11, 11, 256)       0
_________________________________________________________________
flatten (Flatten)            (None, 30976)             0
_________________________________________________________________
dense (Dense)                (None, 1024)              31720448
_________________________________________________________________
dropout_2 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 12)                12300
=================================================================
Total params: 32,121,868
Trainable params: 32,121,228
Non-trainable params: 640
_________________________________________________________________
```

Fig -4: Summary of Proposed CNN Architecture

2.5 Training and Validation Phase

The Table 1 highlights the parameters which are defined in training phase of CNN. In this stage, the training data is used to create the prediction model, whereas the validation data is used to evaluate the performance of the model. The callback functions such as ModelCheckpoint(), ReduceLROnPlateau() and CSVLogger() are used during the training phase of CNN. After completion of the training phase, the prediction model is generated (.h5 file); which is used for making the predictions of biomedical instruments.

Table -1: Training Parameters of CNN

Sl. No.	Parameter	Value
1.	Batch Size	8
2.	Number of Epochs	60
3.	Learning Rate	0.0001
4.	Metric	Accuracy
5.	Optimizer	Adam

The Fig. 5 depicts the details of initial 10 epochs with the accuracy and loss rate during the training and validation phase.

```
Epoch 1/60
45/45 [==============================] - 3s 46ms/step - loss: 9.4433 - accuracy: 0.1921 - val_loss: 5.0253 - val_accuracy: 0.1053

Epoch 00001: val_loss improved from inf to 5.02530, saving model to Model.h5
Epoch 2/60
45/45 [==============================] - 2s 43ms/step - loss: 6.7541 - accuracy: 0.2825 - val_loss: 5.1183 - val_accuracy: 0.0921

Epoch 00002: val_loss did not improve from 5.02530
Epoch 3/60
45/45 [==============================] - 2s 38ms/step - loss: 5.7609 - accuracy: 0.2853 - val_loss: 5.3621 - val_accuracy: 0.1908

Epoch 00003: val_loss did not improve from 5.02530
Epoch 4/60
45/45 [==============================] - 2s 37ms/step - loss: 4.8857 - accuracy: 0.3616 - val_loss: 6.2624 - val_accuracy: 0.0789

Epoch 00004: val_loss did not improve from 5.02530
Epoch 5/60
45/45 [==============================] - 2s 37ms/step - loss: 4.6441 - accuracy: 0.3531 - val_loss: 8.3642 - val_accuracy: 0.1053

Epoch 00005: val_loss did not improve from 5.02530
Epoch 6/60
45/45 [==============================] - 2s 37ms/step - loss: 4.3780 - accuracy: 0.4167 - val_loss: 10.8256 - val_accuracy: 0.0855

Epoch 00006: val_loss did not improve from 5.02530
Epoch 7/60
45/45 [==============================] - 2s 37ms/step - loss: 4.2740 - accuracy: 0.4011 - val_loss: 7.8386 - val_accuracy: 0.1316

Epoch 00007: val_loss did not improve from 5.02530

Epoch 00007: ReduceLROnPlateau reducing learning rate to 1.9999999494757503e-05.
Epoch 8/60
45/45 [==============================] - 2s 39ms/step - loss: 3.9819 - accuracy: 0.5085 - val_loss: 7.5386 - val_accuracy: 0.1645

Epoch 00008: val_loss did not improve from 5.02530
Epoch 9/60
45/45 [==============================] - 2s 39ms/step - loss: 3.9164 - accuracy: 0.5000 - val_loss: 7.0404 - val_accuracy: 0.2368

Epoch 00009: val_loss did not improve from 5.02530
Epoch 10/60
45/45 [==============================] - 2s 37ms/step - loss: 3.8337 - accuracy: 0.5198 - val_loss: 6.2006 - val_accuracy: 0.3158

Epoch 00010: val_loss did not improve from 5.02530
```

Fig -5: Snapshot of Initial 10 Epochs

3. RESULTS

In this study, Google Colab notebook is used to train and evaluate the performance of the model using Python programming language. The Keras and Tensorflow libraries are utilized for CNN training. The performance of the prediction model is evaluated using the accuracy metric.

The Fig. 6 demonstrates the training and validation accuracy of the prediction model for the entire 60 epochs. From the experimental results, training and validation accuracy found is 79.56% and 57.42% respectively.

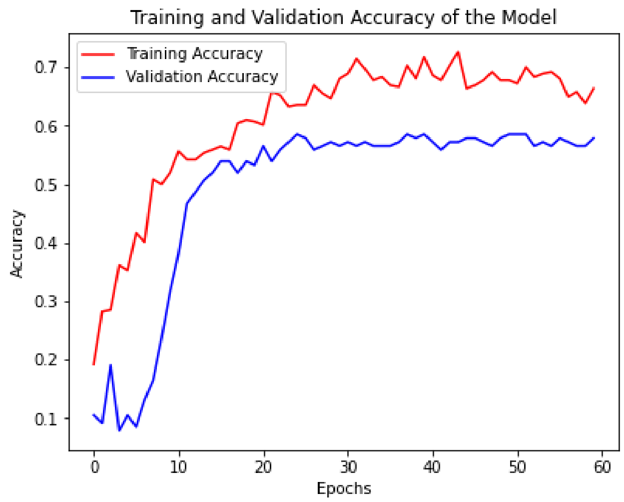

Fig -6: Training and Validation Accuracy of the Model

The Fig. 7 demonstrates the training and validation loss of the prediction model for the entire 60 epochs. From the experimental results, training and validation loss found is 2.9790 and 4.1020 respectively.

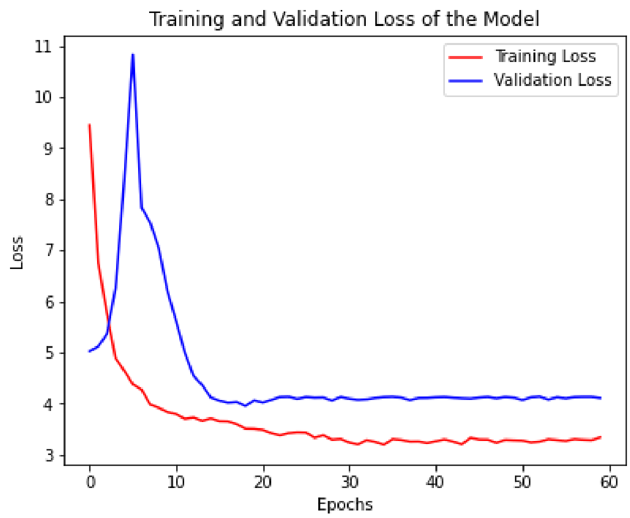

Fig -7: Training and Validation Loss of the Model

The Table 2 summarizes the above experimental results in tabular form.

Table -2: Experimental Results

Sl. No.	Performance Metrics	Training	Validation
1.	Accuracy	79.56%	57.42%
2.	Loss	2.9790	4.1020

The Fig. 8 and Fig. 9 illustrate the classification report of biomedical instruments with respect to training and validation sets respectively.

	precision	recall	f1-score	support
Audiometer	0.13	0.15	0.14	34
CT_scanners	0.04	0.04	0.04	27
Cannula	0.14	0.15	0.14	27
Defibrillator	0.19	0.20	0.20	20
Dialyser	0.13	0.12	0.13	32
Digital_BP_meter	0.15	0.14	0.14	36
Enema_bulb	0.09	0.09	0.09	33
Needle_electrodes	0.05	0.04	0.04	27
Ophthalmoscope	0.04	0.04	0.04	25
Sphygmomanometer	0.17	0.14	0.15	29
Stethoscope	0.00	0.00	0.00	36
Syringe	0.08	0.11	0.09	36
accuracy			0.10	362
macro avg	0.10	0.10	0.10	362
weighted avg	0.10	0.10	0.10	362

Fig -8: Classification Report on Training Set

	precision	recall	f1-score	support
Audiometer	0.18	0.14	0.16	14
CT_scanners	0.10	0.08	0.09	12
Cannula	0.00	0.00	0.00	11
Defibrillator	0.08	0.11	0.09	9
Dialyser	0.00	0.00	0.00	13
Digital_BP_meter	0.17	0.19	0.18	16
Enema_bulb	0.14	0.14	0.14	14
Needle_electrodes	0.00	0.00	0.00	11
Ophthalmoscope	0.00	0.00	0.00	11
Sphygmomanometer	0.00	0.00	0.00	12
Stethoscope	0.19	0.38	0.25	16
Syringe	0.00	0.00	0.00	16
accuracy			0.10	155
macro avg	0.07	0.09	0.08	155
weighted avg	0.08	0.10	0.08	155

Fig -9: Classification Report on Validation Set

4. CONCLUSION

The proposed methodology employed for the identification of biomedical instruments using CNN architecture provides very good results on self prepared dataset. The study can be further extended by including more number of biomedical instruments in the dataset to achieve better results. The proposed methodology could help the robots to identify the biomedical instruments more accurately while performing the medical tasks.

REFERENCES

[1] R.S.Khandpur, Handbook of Biomedical Instrumentation, 2nd Edition, Tata McGraw Hill, 2003.

[2] Joseph J. Carr, Biomedical Equipment- Use, Maintenance & Management, Prentice Hall, 1991.

[3] Lan Good fellow and YoshuaBengio, Deep Learning, MIT Press, 2016.

[4] Sabrina Kletz, Klaus Schoeffmann, Jenny Benois-Pineau, Heinrich Husslein, "Identifying Surgical Instruments in Laparoscopy Using Deep Learning Instance Segmentation", 2019 International Conference on Content-Based Multimedia Indexing (CBMI), pp. 1 – 6, doi:10.1109/CBMI.2019.8877379.

BIOGRAPHIES

Prashant Dhope
M.Tech (VLSI Design and Embedded Systems) Department of E&C Engineering, Center for PG Studies, VTU Belagavi, Karnataka, India – 590 018