Mini Project 2

6th Semester

XX Problem Statement XX

(i) Develop a SoftMax based convolutional neural network for classification of cricket shots.

(ii) Develop a SVM based convolutional neural network for classification of cricket shots.

(iii) Compare the performance of both models using different metrics.

INTRODUCTION

- Cricket is one of the most popular sport in many countries across the world especially in the Indian subcontinent. Batting is a part of the game, which is the art of hitting the ball bowled by bowler in order to score runs.

- Batsmen have to accommodate themselves according to different conditions such as different nature of pitches, different altitudes and different weather. So, for a batsman to be successful, he must, along with better batting skills, have quick reflexes, and be able to adapt to the conditions quickly.

- In recent years, with the growth in the application of Artificial Intelligence, there is a revolution of technology in the field of Cricket also.

INTRODUCTION Contd.

- We have developed Deep Learning models to detect which kind of shot batsman has played.
- Our approach uses convolutional neural networks to classify different types of shots. Our model works on image dataset.
- Our aim is to classify 6 types of cricket shots. We have used the Shot-Net dataset for our project. The developed models will detect the type of cricket shots and apart from that it will also recognize the similarities and differences between different cricket shots.

PROPOSED METHODOLOGY

(A) DATASET

- We have used the ShotNet dataset to train and evaluate our model. It contains 4200 images with 6 classes of cricket shots, namely Straight drive, Pull shot, Cut shot, Cover drive, Leg glance and Scoop shot, with 700 images in each class.

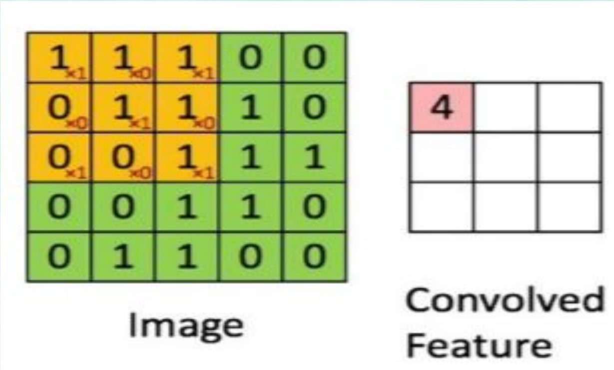
- The authors collected 100 images per class and later augmented it using five different techniques to expand the dataset to 700 images per class.

- The augmentation techniques used were Rotation by $\pm 30^{\circ}$, Shearing, Addition of salt and pepper noise, translation and shading.

(B) BACKGROUND OF CNN

- This is an artificial neural network also known as ANN feedforward. A filter or moving centre runs through the picture in the convolutional layer. It usually occupies a specific part of a 2D matrix (image representation), applies point multiplication, and stores the result in another matrix.
- Convolution is represented by following mathematical formula, (the size of filter is $(2a + 1) \times (2b + 1)$) :

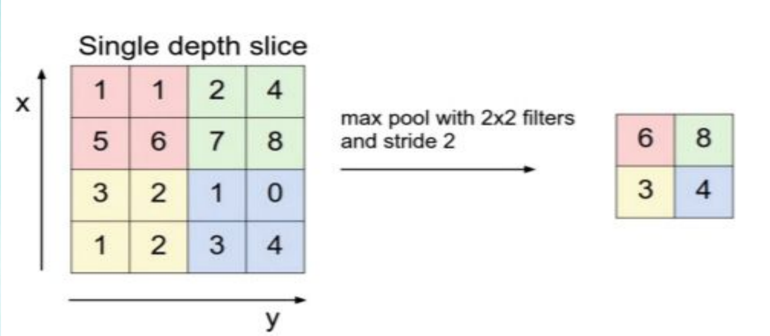
$$h(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} F(s, t) I(x + s, y + t)$$



Convolution of an image with a filter

BACKGROUND OF CNN Contd.

- The pooling layer is generally adjacent to the convolution layer. It was mainly used for memory reduction and quick calculation. Decreases the volume. Max pooling is one of the most used levels by CNN.



Max pooling with 2×2 filter with stride 2

BACKGROUND OF CNN Contd.

- A flatten layer converts 2D or 3D array input from previous layer converts it to 1D array. A fully connected layer is the one where all the outputs of previous layer is connected to every neuron of the layer. The output layer of the neural network shows the probability of the classes. It is calculated by the "SoftMax" function. The equation for calculating the probability is given below:

$$\sigma(X_i) = \frac{e^{X_i}}{\sum_j e^{X_j}}$$

(C) BACKGROUND OF SVM

i. Support Vector Machines

Given training data and its corresponding labels. SVMs learning consists of the following constrained optimization:

$$\min_{w,\xi_n} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{n=1}^{N}\xi_n$$

$$s.t.\ \boldsymbol{w}^T\boldsymbol{x}_n t_n \geq 1 - \xi_n\ \forall n$$

$$\xi_n \geq 0\ \forall n$$

ξ_n are slack variables which penalizes data points which violate the margin requirements. The corresponding unconstrained optimization problem is:

$$\min_{w,} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{n=1}^{N}\max\left(1 - \boldsymbol{w}^T\boldsymbol{x}_n t_n, 0\right)$$

BACKGROUND OF SVM Contd.

ii. Multiclass SVM

For k class problems, k linear SVMs will be trained independently, where the data from the other classes form the negative cases. The output of the k SVM is:

$$a_k(\boldsymbol{x}) = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}$$

The predicted class is: $\arg\max_{k} a_k(\boldsymbol{x})$

Loss Functions

1. Categorical cross entropy for CNN-SoftMax model:

$$L_{CE} = -\sum_{i=1}^{n} t_i \log(p_i)$$

where *ti* is the truth label and *pi* is the SoftMax probability for the *ith* class.

2. Square hinge loss for CNN-SVM model:

$$L_{SH} = \sum_{i=0}^{n} (\max(0,1 - y.y')^2)$$

where ŷ the predicted value and y is either 1 or -1.

PROPOSED MODEL

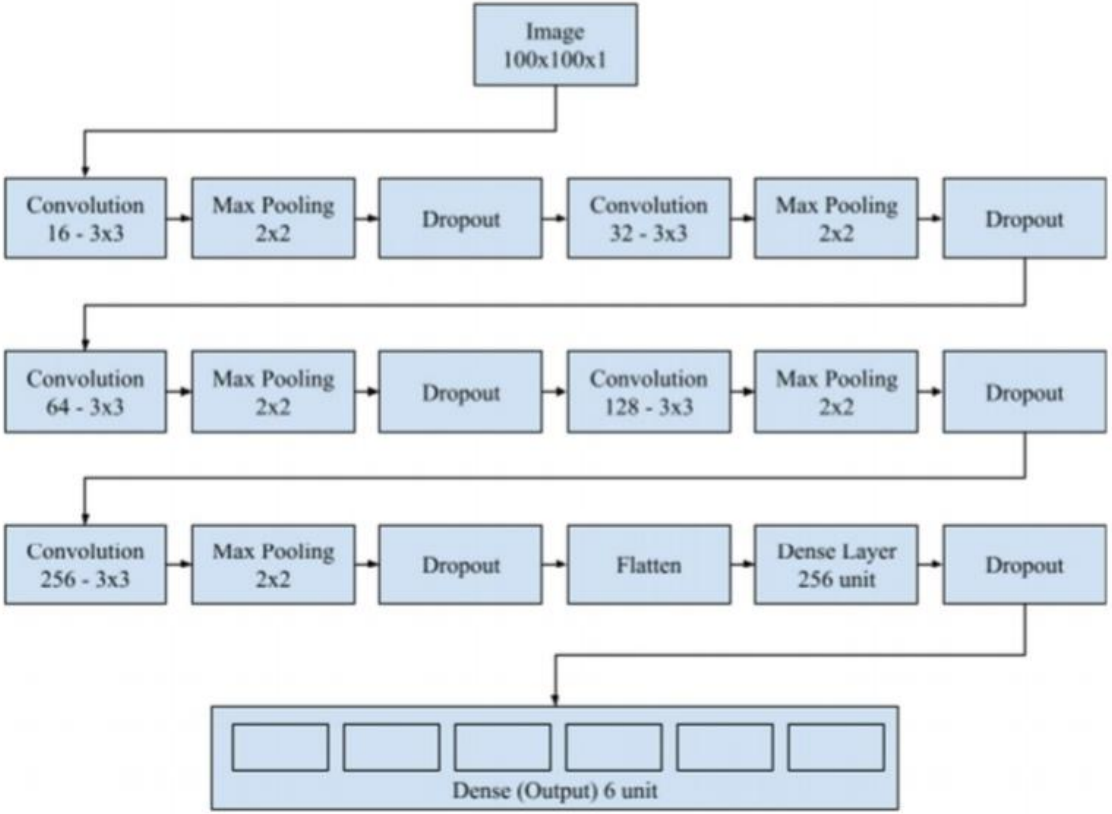


Figure 4: Architecture of our model

TRAINING THE MODEL

- We have used “Adam optimizer” to compile both the models. 80% of the dataset is used training and 20% for testing.

- The training dataset has 3360 images with 540 images per shot. The test dataset has 840 images with 140 images per shot.

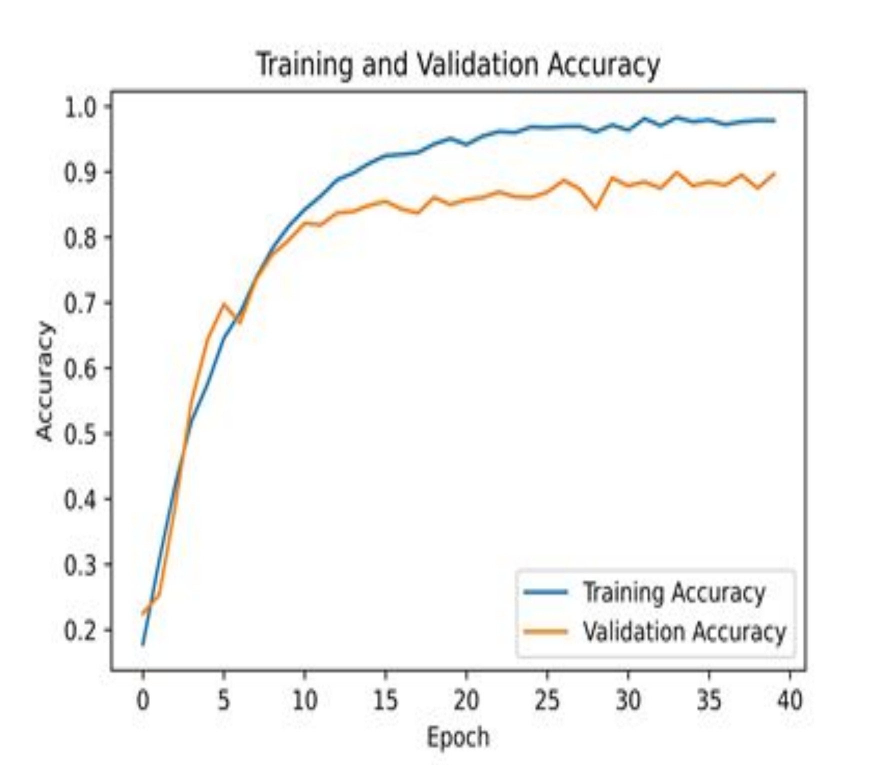
- 5-Fold cross validation is used for evaluating the performance. The dataset is split into five equal parts, the model is trained on four parts, and validated on the remaining one.

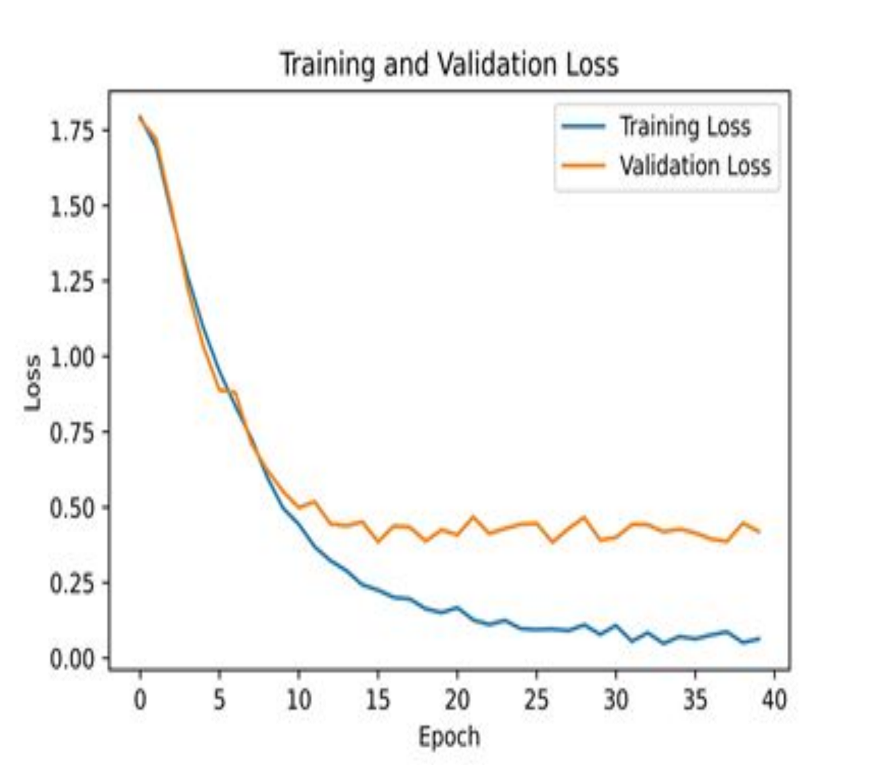
- This is repeated five times with every part used as validation set exactly once. The network was trained for 40 epochs in every fold.

EXPERIMENTAL RESULTS

It can be observed that CNN-SoftMax model converged with a good validation accuracy at low cross entropy loss. CNN-SVM model also converged with a fairly high accuracy and low squared hinge loss.

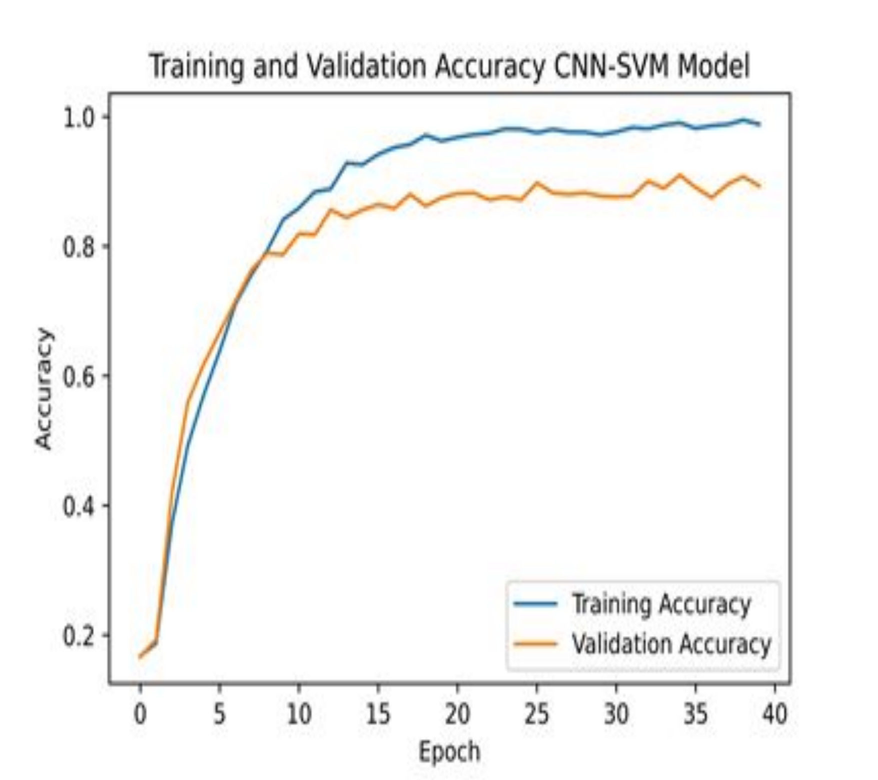
A) TRAINING PHASE

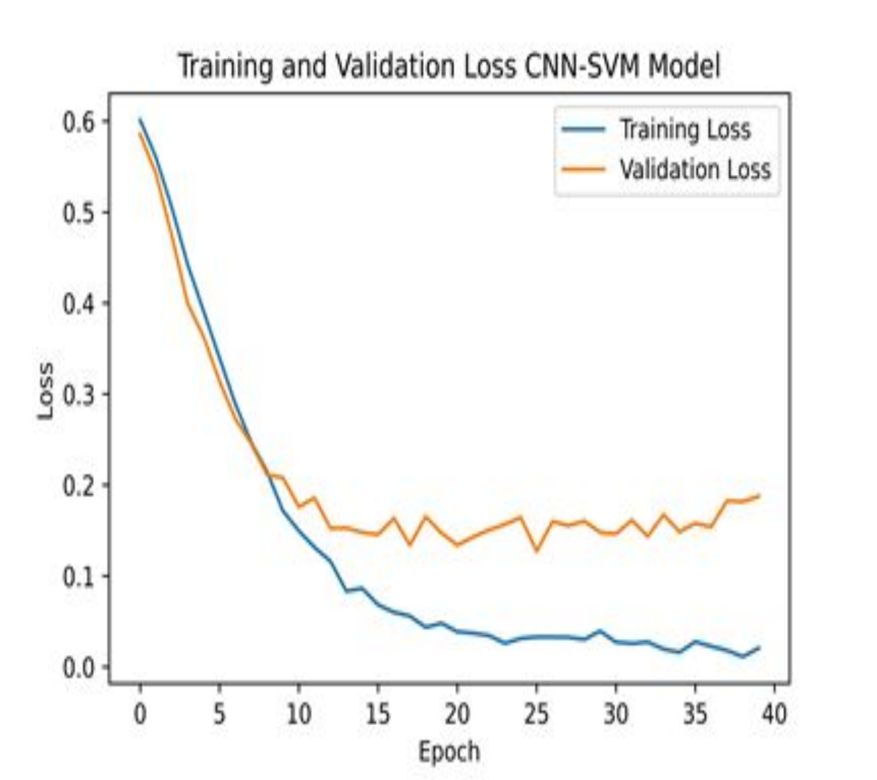




CNN-SoftMax model converged with a good validation accuracy at low cross entropy loss.

A) TRAINING PHASE Contd





CNN-SVM model also converged with a fairly high accuracy and low squared hinge loss.

B) TESTING PHASE

- We evaluated the trained model on the following metrics: precision, F1-score and recall.

- It can also be observed that both the models beat Shot-Net model in performance by a fair margin (80% accuracy in Shot-Net and around 90% in our models).

Table 1: Classification report of CNN-SoftMax Model

Shot	Precision	Recall	F1-Score
Cut Shot	0.8943	0.9071	0.9007
Cover Drive	0.9136	0.9071	0.9103
Straight Drive	0.8851	0.9357	0.9097
Pull Shot	0.9357	0.9357	0.9357
Leg Glance	0.9624	0.9142	0.9377
Scoop Shot	0.9347	0.9214	0.9280
Average	0.9210	0.9202	0.9203

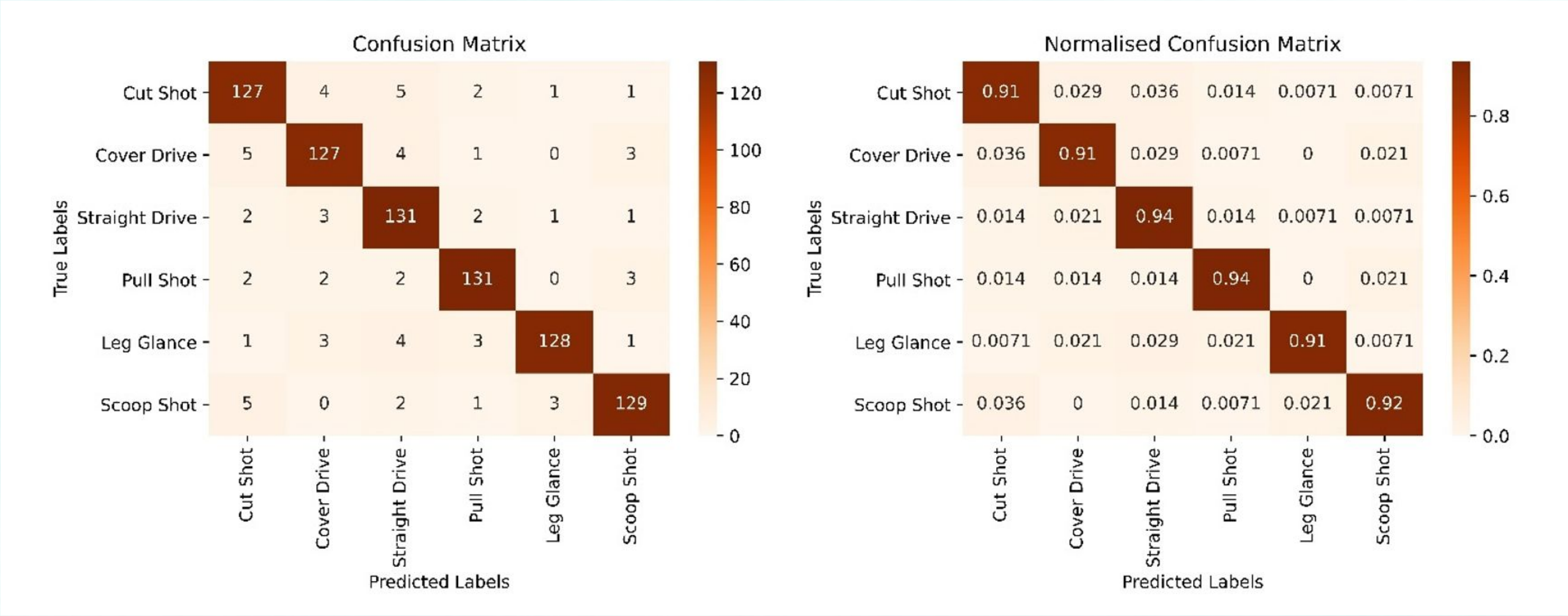
Table 2: Classification report of CNN-SVM Model

Shot	Precision	Recall	F1-Score
Cut Shot	0.7975	0.9285	0.8580
Cover Drive	0.8881	0.9071	0.8975
Straight Drive	0.9612	0.8857	0.9219
Pull Shot	0.9275	0.9142	0.9208
Leg Glance	0.9603	0.8642	0.9097
Scoop Shot	0.9078	0.9142	0.9110
Average	0.9070	0.9023	0.9032

Table 3: Classification report of Shot-Net Model [11]

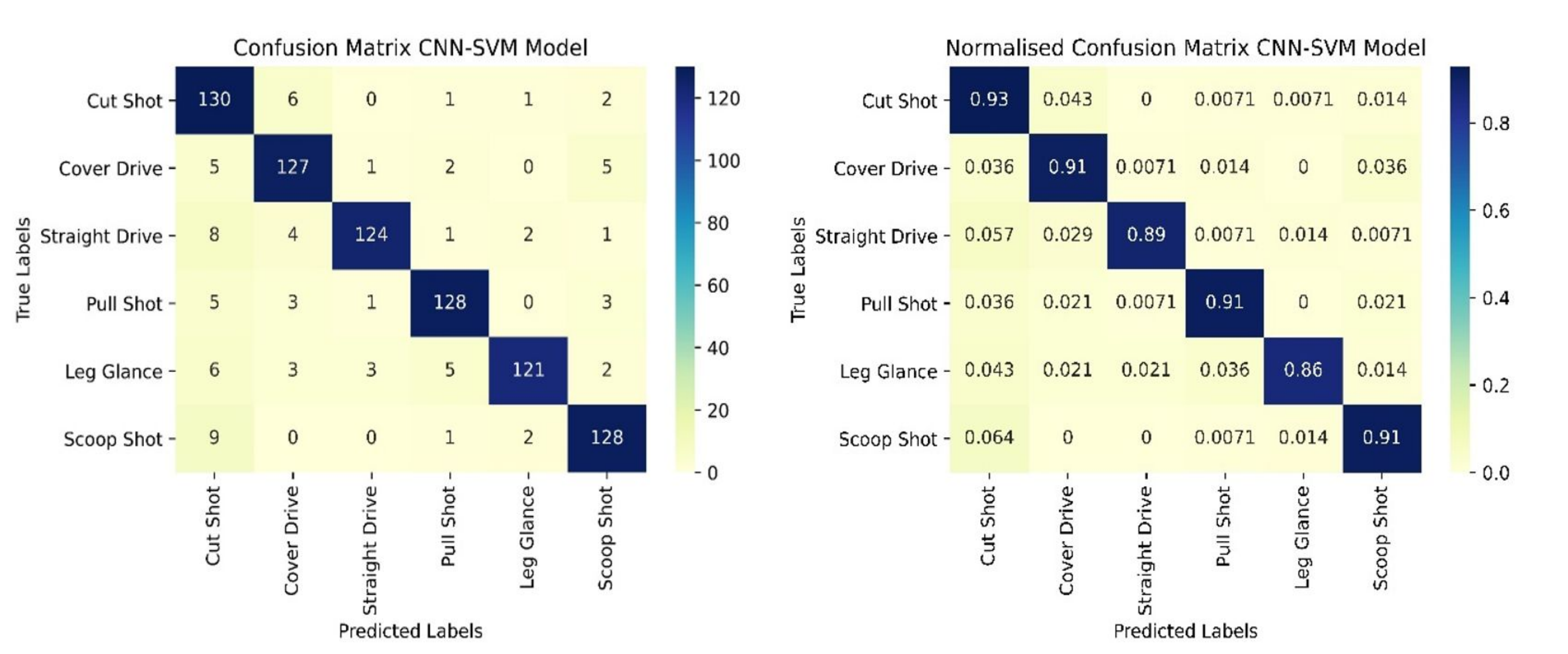
Shot	Precision	Recall	F1-Score
Cut Shot	0.69	0.78	0.72
Cover Drive	0.74	0.78	0.76
Straight Drive	0.78	0.83	0.81
Pull Shot	0.89	0.77	0.83
Leg Glance	0.79	0.88	0.83
Scoop Shot	0.88	0.72	0.79
Average	0.80	0.79	0.79

B) TESTING PHASE Contd.



CNN-SoftMax Model (a) confusion matrix (L) (b) normalised confusion matrix (R)

B) TESTING PHASE Contd.



CNN-SVM Model (a) confusion matrix (L) (b) normalised confusion matrix (R)

C) K-FOLD CROSS VALIDATION RESULTS

- k-fold cross validation is a rigorous method to evaluate the performance of the model.

Table 4: 5-Fold cross validation results

	Accuracy	
Fold	**CNN-SoftMax Model**	**CNN-SVM Model**
1	0.8917	0.8833
2	0.8988	0.9000
3	0.8917	0.8929
4	0.9036	0.8976
5	0.8857	0.9143
Avg.	0.8943	0.8976

CONCLUSION

- In this study, we have provided two models for classification of cricket shots using CNN-SoftMax model and CNN-SVM model (Shot-Net Pro) and compared the results achieved from both of them.

- We used five convolution layers, five max pooling layers, six dropout layers, one flatten layer and two dense layers. The final achieved result we've found from both the two is quite promising.

- CNN-SVM performed slightly better than CNN-SoftMax.

- Both the models outperformed the Shot-Net model by a fair margin of 10%. We hope that these methods will be developed as real applications in the future for the well being of the game of cricket.

FUTURE WORK

- In future the network can be upgraded to create a better neural network for better accuracy.
- We can upgrade this network to perform image classification on RGB images.
- The model can also be modified to perform classification on video dataset so that commentary can be automated in future.
- We can perform classification based on 3D depth images using deep learning. We can use different types of algorithms choosing which one is efficient.

Project done by :

- Md.Ezaz Anwar IIT2018004
- Kalyan Kumar Dubey IIT2018005
- Ankit Kumar Das IIT2018020
- Raushan Raj IIT2018031

Thank You