Vision-Based Multi-Modal Framework for Action Recognition

Team - 38
Sarthak Jain 191IT145
Yash Gupta 191IT158
Rishit 191IT141

Introduction

Human activity recognition is crucial in the development of intelligent systems for video surveillance, public security, health care, and home monitoring, where activity detection and recognition can improve human security and quality of life.

In order to avoid unsafe circumstances, automated, intuitive, and real-time technologies are necessary to understand human actions and accurately identify odd behaviours.

We hope to use this research to investigate how two modalities (RGB and skeletal data) can be combined to create a strong multi-modal framework for vision-based human activity detection.

Finally, Illustrative representations derived from a combination of dynamic RGB images, and skeleton data representations will be used to highlight spatial information, body shape/posture, and temporal evolution of actions.

Problem Statement

Recognizing Human Action using Vision-Based Multi-Modal Framework based on RGB and skeleton data

Literature

M. Haghighat, M. Abdel-Mottaleb, and W. Alhalabi, “Fully automatic face normalization and single sample face recognition in unconstrained environments,” Expert Systems with Applications, vol. 47, pp. 23–34,2016 :-We learnt about the methods for recognizing faces from non-frontal views and under different illumination conditions using only a single gallery sample for each subject.

G. Chetty and M. White, “Body sensor networks for human activity recognition,” in 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN). IEEE, 2016, pp. 660–665 :- We gained knowledge regarding activity recognition for identity verification, health and ageing, and sports and exercise monitoring applications.

P. Khaire, J. Imran, and P. Kumar, “Human activity recognition by fusion of rgb, depth, and skeletal data,” in Proceedings of 2nd International Conference on Computer Vision & Image Processing. Springer, 2018, pp. 409–421 - We discovered effective, quickly trainable and customizable system for recognizing human activities designed with an automated machine learning method based on Neural network.

Objectives

1)Dynamic image construction for RGB and retrieving skelton images based on joint angle and location.

2)Transfer learning for feature extraction from above generated images.

3)Combining features from generated images.

4)Applying dimensionality reduction techniques on the data.

5)Training various ML and DL algorithms on the data for classifying the human action.

6)Also comparison is made between various modalities and based on different feature extractors

Datasets

UTD-MHAD dataset :

UTD-MHAD dataset consists of 27 different actions: (1) right arm swipe to the left, (2) right arm swipe to the right, (3) right hand wave, (4) two hand front clap, (5) right arm throw, (6) cross arms in the chest, (7) basketball shoot, (8) right hand draw x, (9) right hand draw circle (clockwise), (10) right hand draw circle (counter clockwise), (11) draw triangle, (12) bowling (right hand), (13) front boxing, (14) baseball swing from right, (15) tennis right hand forehand swing, (16) arm curl (two arms), (17) tennis serve, (18) two hand push, (19) right hand knock on door, (20) right hand catch an object, (21) right hand pick up and throw, (22) jogging in place, (23) walking in place, (24) sit to stand, (25) stand to sit, (26) forward lunge (left foot forward), (27) squat (two arms stretch out).

Data is collected using Microsoft Kinect sensor and a wearable inertial sensor.The dataset contains 27 actions performed by 8 subjects (4 females and 4 males).Each subject repeat each action 4 times.

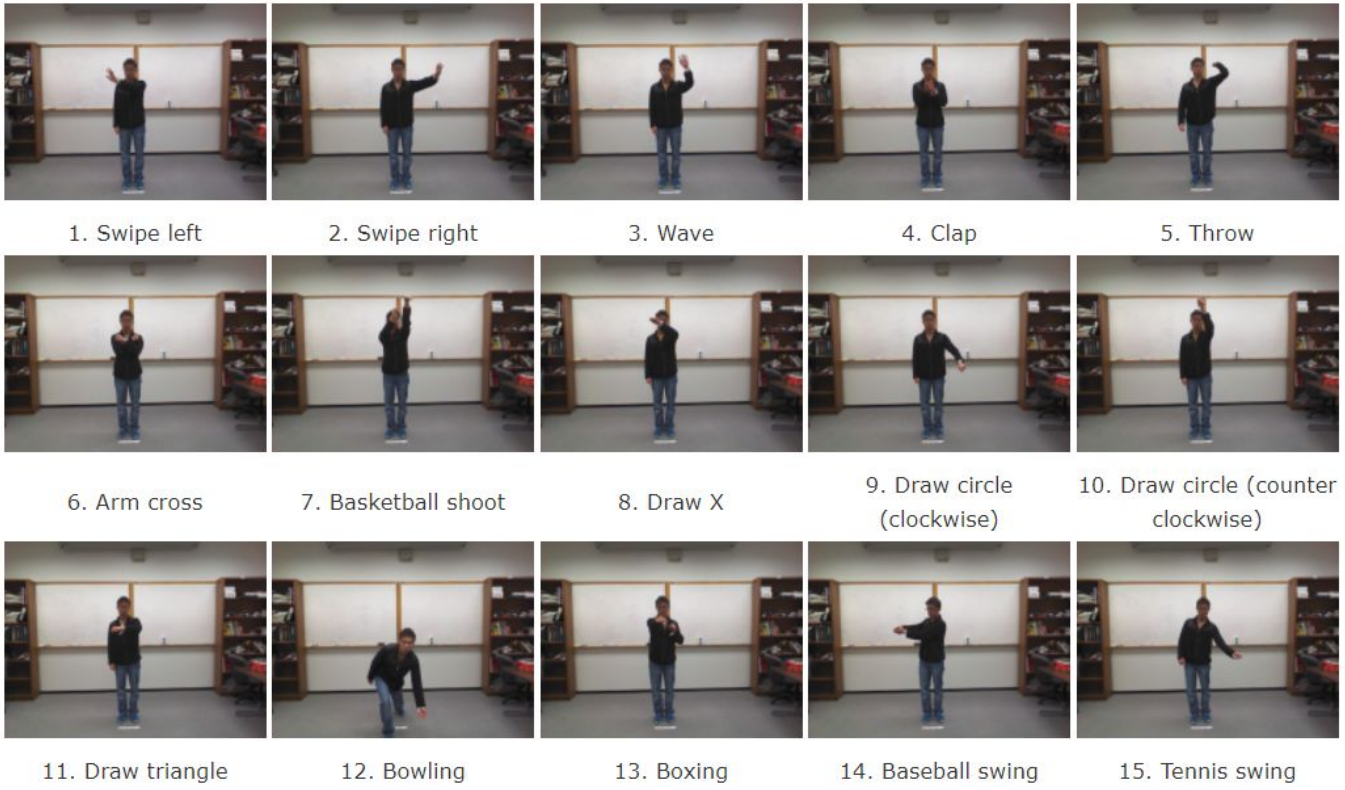
Datasets

A Kinect camera is used to obtain the dataset.

Videos are recorded at 30 frames per second. All above mentioned 27 actions are performed by 8 subjects, 4 males and 4 females. Each action is performed by subjects 4 times.

Total data points in the dataset are 864.

Actions Recognized



16. Arm curl

17. Tennis serve

18. Push

19. Knock

20. Catch

21. Pickup and throw

22. Jog

23. Walk

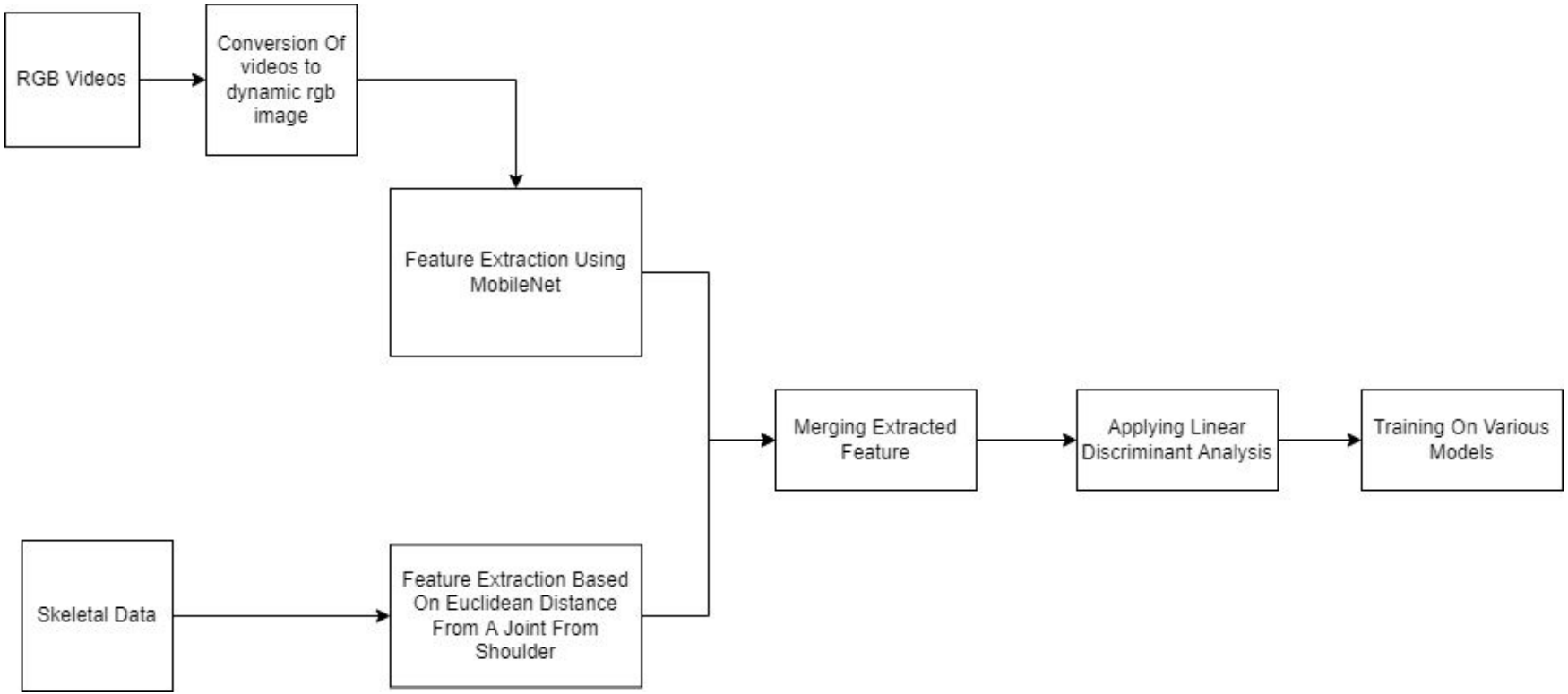
24. Sit to stand

25. Stand to sit

26. Lunge

27. Squat

Model Diagram



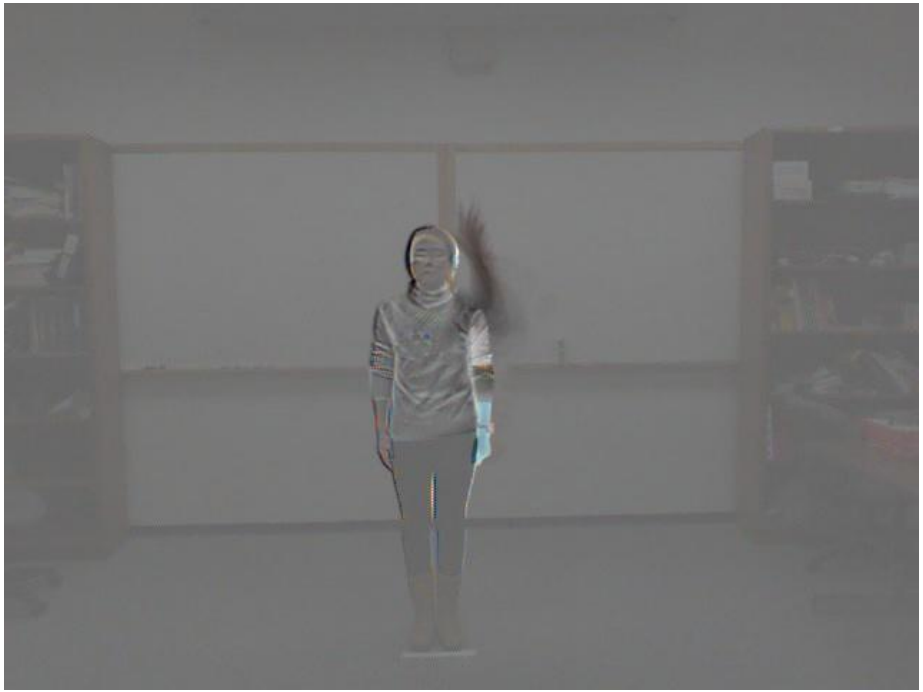
Methodology

1)Conversion of RGB videos to Dynamic RGB image :

```
for n in range(num_frames):
    cumulative_indices = np.array(range(n, num_frames)) + 1
    print('cumulative_indice : ',cumulative_indices)
    coefficients[n] = np.sum(((2*cumulative_indices) - num_frames) / cumulative_indices)
    print('coefficients[n] : ',n,coefficients[n])
```

Multiply the calculated coefficients with the frames and than find the cumulative sum

Dynamic RGB images :



- Features are extracted from dynamic RGB image using Mobilenet.
- Features are extracted from skeletal data based on relative distance between the joints.
- Extracted features are merged.
- Dimensionality Reduction techniques such as LDA (Linear Discriminant Analysis) are used.
- Various models are used on the final data and best model is chosen.

Checking on single modality(RGB)

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
mobilenet_1.00_224 (Function (None, 8, 8, 1024)        3228864
_________________________________________________________________
global_average_pooling2d_1 ( (None, 1024)              0
_________________________________________________________________
dense_9 (Dense)              (None, 27)                27675
=================================================================
Total params: 3,256,539
Trainable params: 27,675
Non-trainable params: 3,228,864
```

Network Trained on extracted features

Layer (type)	Output Shape	Param #
dense_18 (Dense)	(None, 200)	5200
dropout_24 (Dropout)	(None, 200)	0
dense_19 (Dense)	(None, 150)	30150
dropout_25 (Dropout)	(None, 150)	0
dense_20 (Dense)	(None, 100)	15100
dropout_26 (Dropout)	(None, 100)	0
dense_21 (Dense)	(None, 27)	2727

Total params: 53,177
Trainable params: 53,177
Non-trainable params: 0

Results :

1) On Dynamic Image RGB:

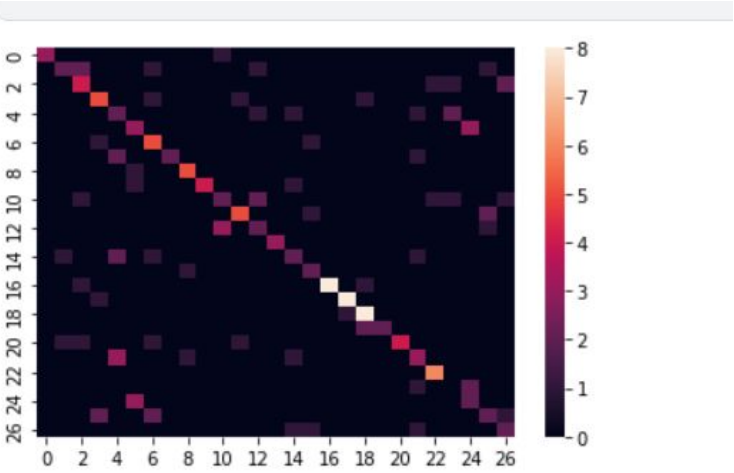
```
]:
# y_test = pd.get_dummies(y_test).to_numpy()
y_tes = pd.get_dummies(y_test).to_numpy()
base_model.evaluate(np.array(X_test), np.array(y_tes),verbose=0)[1]
```

```
]: 0.7398843765258789
```

2)On skeletal data:

```
dtc = DecisionTreeClassifier()
dtc.fit(X_train1, y_train1)
y_pred = dtc.predict(X_test1)
print("Accuracy: ",accuracy_score(y_test1,y_pred))
```

```
Accuracy:  0.5549132947976878
```



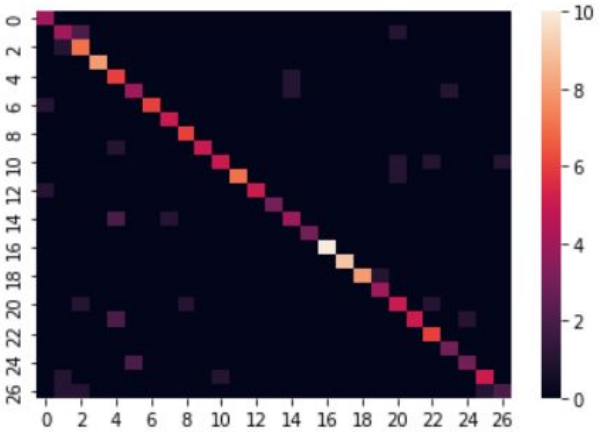
[69]:

```
import lightgbm as lgb
dtc = lgb.LGBMClassifier()
dtc.fit(X_train1, y_train1)
y_pred = dtc.predict(X_test1)
print("Accuracy: ",accuracy_score(y_test1,y_pred))
```

```
Accuracy:  0.8208092485549133
```

[70]:

```
sns.heatmap(confusion_matrix(y_test1,y_pred))
plt.show()
```

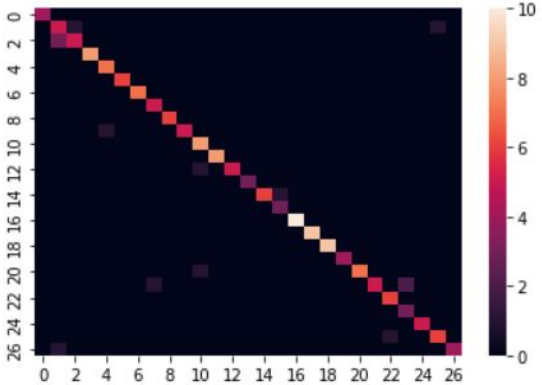


3)On MultiModal Combination:

[71]:
```
dtc = ExtraTreesClassifier()
dtc.fit(X_r2, y_train1)
y_pred = dtc.predict(X_r3)
print("Accuracy: ",accuracy_score(y_test1,y_pred))
print("Accuracy: ",accuracy_score(y_test1,y_pred))
```

```
Accuracy:  0.9190751445086706
Accuracy:  0.9190751445086706
```

```
sns.heatmap(confusion_matrix(y_test1,y_pred))
plt.show()
```



Comparison of different extractors + models

Model	Accuracy
Model 1 (with MobileNet as feature extractor) On dynamicRGB data	70.5% on test data 91.5% during training
Model 2 (Decision Tree) on skeltal data	60.6%
Model 3 (LGBMClassifier) on skeltal data	82.08%
Model 4 (decision tree on multi modal combination)	56%
Model 5 (lightGBM on multi modal combination)	85.5%
Model 6 (XGBClassifier on multi modal combination)	78%
After performing dimensionality reduction using PCA lightGBM gives best accuracy	73.44%
After performing dimensionality reduction using LDA ExtraTreeClassifier gives best accuracy	93.06%
Deep Neural Network on fused multimodal features	85.5%
Best accuracy when VGG Extractor is used is obtained on extra tree classifier	91.3%

Stacked Model

```
def get_stacking():
    level0 = list()
    level0.append(('tree1',ExtraTreesClassifier()))
    level0.append(('tree2',ExtraTreesClassifier()))
    level0.append(('tree3',ExtraTreesClassifier()))
    level0.append(('tree4',ExtraTreesClassifier()))
    level1 = ExtraTreesClassifier()
    model = StackingClassifier(estimators=level0, final_estimator=level1, cv=5)
    return model
```

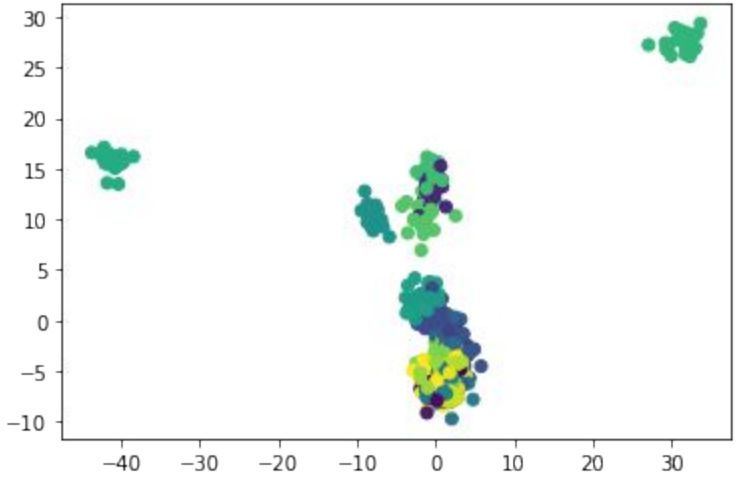
Accuracy Scores

```
y_pred = model.predict(X_r3)
print("Accuracy: ",accuracy_score(y_test1,y_pred))
```

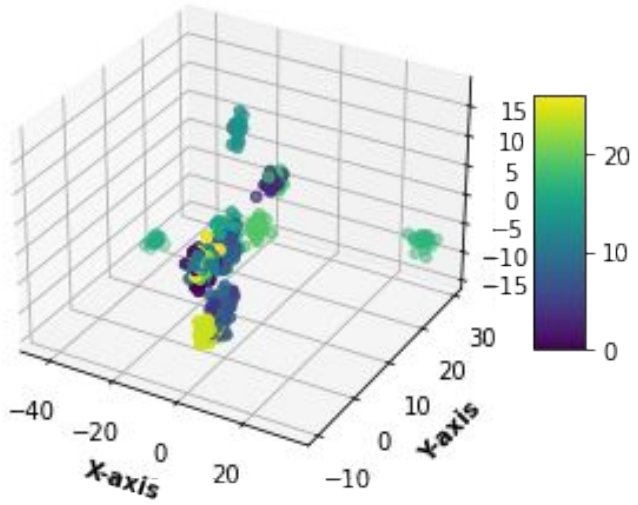
```
Accuracy:  0.9017341040462428
```

Accuracy on Test Set

Visualisation:



When
n_components=2



When
n_components=3

Innovations

1)Fusion of RGB based modalarity and skeletal based (where features are extracted on the based on Euclidean space vector)

2)Comparitive study by using multiple methods which includes stacking,and also different methods for dimensionality reductions like Linear Discriminant Analysis etc. on feature fusion vector.

Thank You