HUMAN ACTION RECOGNITION

(USING 3D CNN)

BY-ABHINAV KUMAR

SUPERVISOR: Dr. Parashjyoti Borah

AIM

Recognizing different human actions from images or videos.

EXAMPLE:

BOXING

HANDCLAPPING

RUNNING

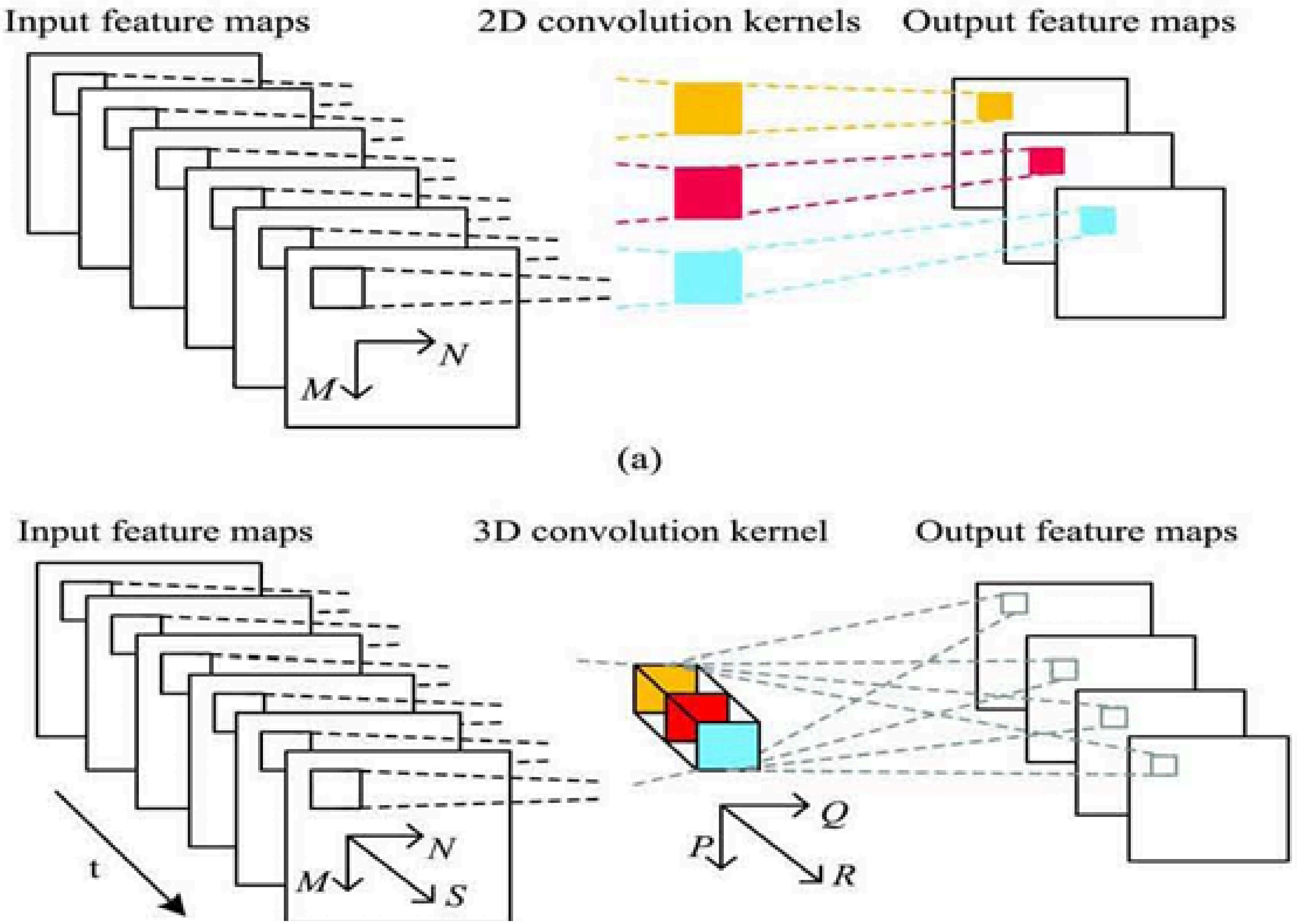
WALKING

WHY DO WE NEED THIS?

- Remote Patient Monitoring
- Elderly Care
- Fitness Tracking
- Safety Monitoring
- Surveillance and Security

  etc...

WHY 3D CNN?



Input feature maps
2D convolution kernels
Output feature maps
M
N
(a)
Input feature maps
3D convolution kernel
Output feature maps
t
M
N
S
P
Q
R

3D CNN model uses two CNN streams:

SPATIAL STREAM

- It process individual RGB frame to capture object and scene appearance.

TEMPORAL STREAM

- It uses optical flow between frames to capture motion information

NOTE:2D CNNs operate on individual frames or images, treating each frame independently. They excel at extracting spatial features (e.g., shapes, textures) but fail to capture the temporal relationships

METHODOLOGY

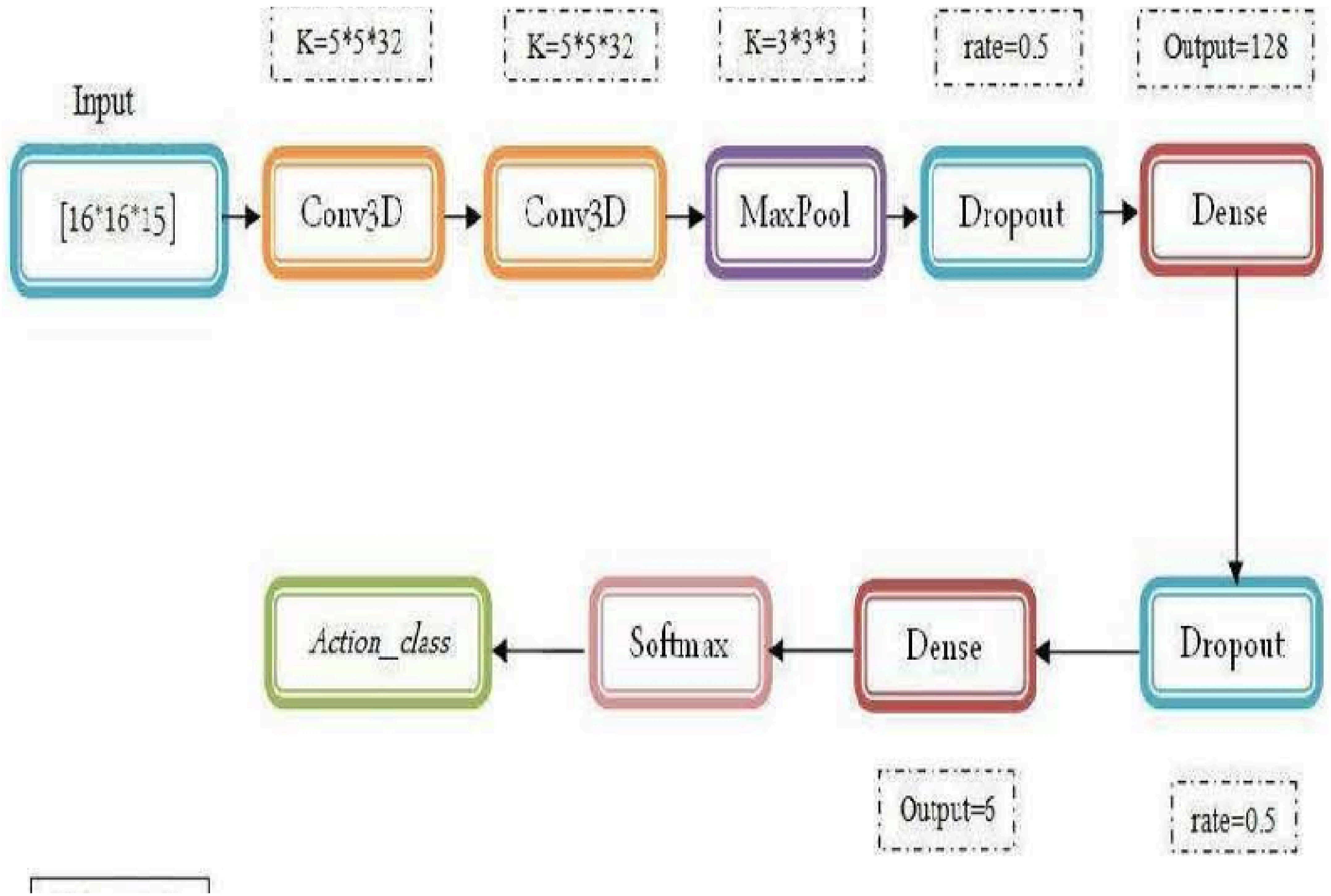
DATASET

- KTH
- JHMDB
- HMDB
- CAVIAR

PREPROCESSING

- **Extract_frames**:Divide the videos into frames
- **OpenCV**: converts the color space from BGR to RGB
- **Normalization**: Pixel values are normalized to improve model performance.

WORKING


K=5*5*32
K=5*5*32
K=3*3*3
rate=0.5
Output=128
Input
[16*16*15]
Conv3D
Conv3D
MaxPool
Dropout
Dense
Action_class
Softmax
Dense
Dropout
Output=6
rate=0.5

TOOLS

- **TENSERFLOW** : A deep learning framework for building and training neural networks
- **MAXPOOLING**: to reduce the dimentionality of image
- **DROPOUT**: prevent overfitting
- **SOFTMAX**: converts the raw output scores from the model into probabilities that sum to 1.
- **Dense:** Fully connected layer for classification.
- **Flatten:** Flattens the input for feeding into dense layers
- **K-Fold Cross-Validation:**model's performance is evaluated robustly and that the results are not biased by a specific train-test split.

LOSS FUNCTION

CATEGORICAL CROSS ENTROPY

$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output}\\\text{size}}} y_i \cdot \log \hat{y}_i$$

- **yi:** The true label (ground truth) for class i
- **yi(cap):** is the predicted probability for class i
- **n:** The total number of classes.

IMPLEMENTATION AND PERFORMANCE

- implemented basic code ..

```
--- Fold 1 Performance ---
Accuracy: 0.8833
Confusion Matrix:
[[23  0  0  0  0  0]
 [ 0 18  1  0  0  0]
 [ 0  1 19  0  0  0]
 [ 0  0  0 12  2  6]
 [ 0  0  0  1 14  0]
 [ 0  1  0  2  0 20]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        23
           1       0.90      0.95      0.92        19
           2       0.95      0.95      0.95        20
           3       0.80      0.60      0.69        20
           4       0.88      0.93      0.90        15
           5       0.77      0.87      0.82        23

    accuracy                           0.88       120
   macro avg       0.88      0.88      0.88       120
weighted avg       0.88      0.88      0.88       120
```

Fold1 performance

```
--- Fold 2 Performance ---
Accuracy: 0.8167
Confusion Matrix:
[[21  1  0  0  0  0]
 [ 0 14  3  0  0  1]
 [ 1  0 21  0  0  0]
 [ 1  0  0 12  5  0]
 [ 0  0  0  6 14  0]
 [ 0  0  1  3  0 16]]
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.95      0.93        22
           1       0.93      0.78      0.85        18
           2       0.84      0.95      0.89        22
           3       0.57      0.67      0.62        18
           4       0.74      0.70      0.72        20
           5       0.94      0.80      0.86        20

    accuracy                           0.82       120
   macro avg       0.82      0.81      0.81       120
weighted avg       0.83      0.82      0.82       120
```

Fold2 performance

```
--- Fold 3 Performance ---
Accuracy: 0.8083
Confusion Matrix:
[[19  1  0  0  0  0]
 [ 0 20  1  0  0  0]
 [ 1  1 11  0  0  0]
 [ 0  0  0 15  3  8]
 [ 1  0  1  2 17  1]
 [ 0  0  0  3  0 15]]
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.95      0.93        20
           1       0.91      0.95      0.93        21
           2       0.85      0.85      0.85        13
           3       0.75      0.58      0.65        26
           4       0.85      0.77      0.81        22
           5       0.62      0.83      0.71        18

    accuracy                           0.81       120
   macro avg       0.81      0.82      0.81       120
weighted avg       0.81      0.81      0.81       120
```

Fold3 performance

```
--- Fold 4 Performance ---
Accuracy: 0.8151
Confusion Matrix:
[[11  1  1  1  0  0]
 [ 1 19  2  0  0  0]
 [ 0  1 21  0  0  0]
 [ 0  0  0 13  3  2]
 [ 0  0  0  5 14  1]
 [ 1  0  0  3  0 19]]
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.79      0.81        14
           1       0.90      0.86      0.88        22
           2       0.88      0.95      0.91        22
           3       0.59      0.72      0.65        18
           4       0.82      0.70      0.76        20
           5       0.86      0.83      0.84        23

    accuracy                           0.82       119
   macro avg       0.82      0.81      0.81       119
weighted avg       0.82      0.82      0.82       119
```

Fold4 performance

```
--- Fold 5 Performance ---
Accuracy: 0.7731
Confusion Matrix:
[[16  4  0  0  0  0]
 [ 2 16  0  0  0  1]
 [ 1  2 20  0  0  0]
 [ 0  0  0 12  1  5]
 [ 0  0  0  9 13  1]
 [ 0  0  0  1  0 15]]
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.80      0.82        20
           1       0.73      0.84      0.78        19
           2       1.00      0.87      0.93        23
           3       0.55      0.67      0.60        18
           4       0.93      0.57      0.70        23
           5       0.68      0.94      0.79        16

    accuracy                           0.77       119
   macro avg       0.79      0.78      0.77       119
weighted avg       0.80      0.77      0.78       119
```

Fold5 performance

```
=== Final Performance (Averaged Over All Folds) ===
Average Accuracy: 0.8193
Average Precision: 0.8248
Average Recall: 0.8206
Average F1-Score: 0.8172
```

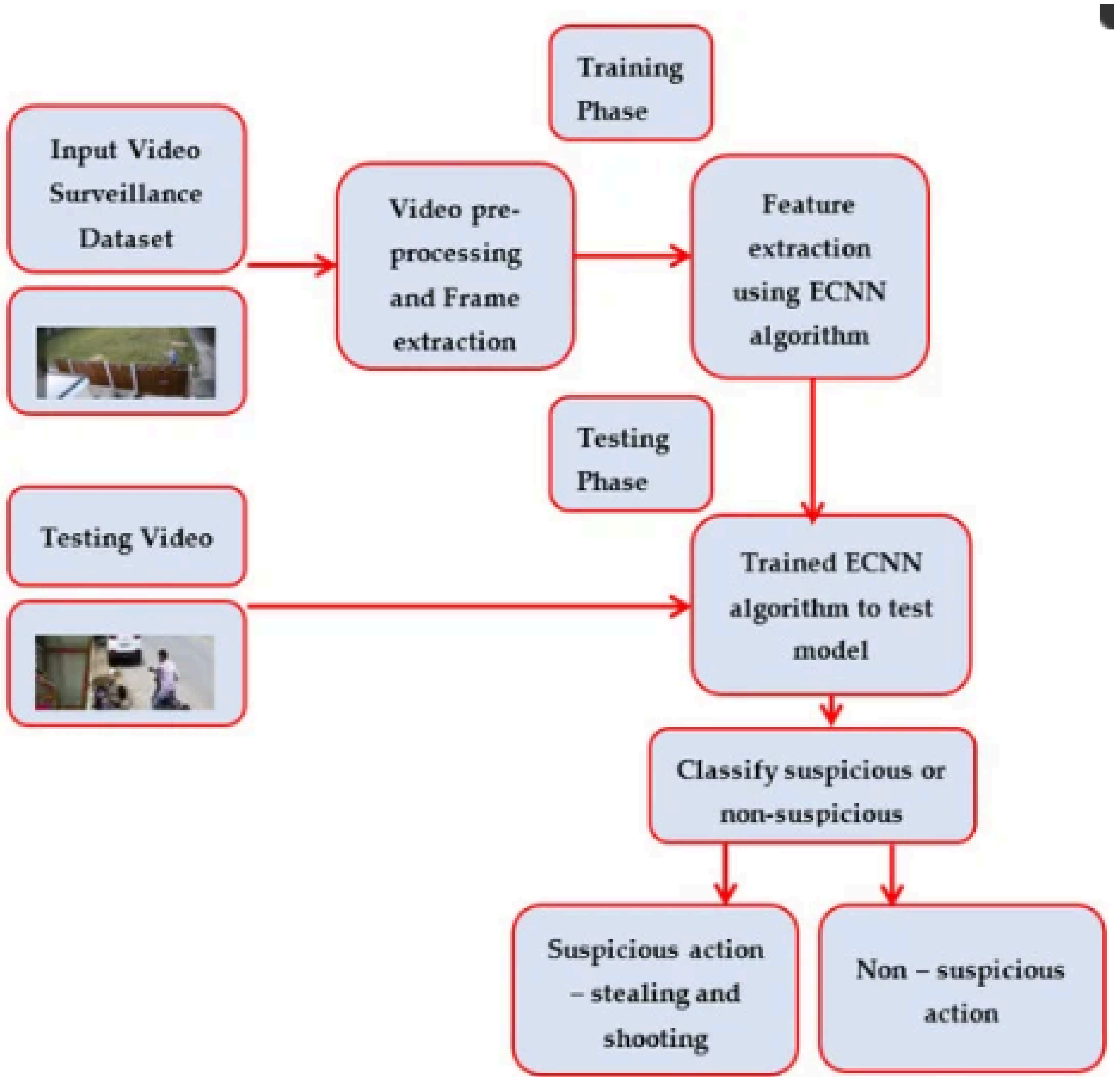
Final performance(average)

FUTURE CHALLENGES

- Preprocessing video data for 3D CNNs is **more complex** than for 2D CNNs.
  Example: Resizing video frames to a fixed resolution
- 3D CNNs are **computationally expensive**, making real-time action recognition difficult.
  Example: surveillance require low-latency which is challenging.
- **Hardware Limitations**
  Challenge: Training and deploying 3D CNNs require powerful hardware (e.g., GPUs or TPUs)

FUTURE WORK

- exploring(3D CNN + LSTM)
- Suspicious Activity Detection from Surveillance Video



REFERENCES

- https://ieeexplore.ieee.org/abstract/document/9429429
- https://ieeexplore.ieee.org/abstract/document/8285700
- https://ieeexplore.ieee.org/abstract/document/9074920
- image sources: google.com

THANK YOU