Ex No	Model	Hyper-parameters	Result	Decision + Explanation
Experiments with Conv3D CNN models				
1	3 Conv3D layers with filters [32, 64, 128]	**Image Size**: 128x128 - **Frames**: 15 (Alternate frames) - **Batch Size**: 32 - **Epochs**: 50 - **Learning Rate**: 0.0001 - **Optimizer**: Adam	Training accuracy: 0.53 Validation Accuracy: 0.47	- **Conv3D Layers**: Three layers capture spatial-temporal features progressively. - **Filter Sizes**: Increasing filters (32→64→128) for learning complex patterns. - **Image Size**: 128x128 balances detail and computational load. - **Frames Considered**: Alternate frames reduce redundancy and computation. - **Learning Rate**: Low rate for gradual convergence.
2	4 Conv3D layers with filters [32, 64, 128, 256]	- **Image Size**: 128x128 - **Frames**: 30 (All frames) - **Batch Size**: 32 - **Epochs**: 50 - **Learning Rate**: 0.001 - **Optimizer**: Adam	Training accuracy: 0.98 Validation Accuracy: 0.65	**Additional Conv3D Layer**: Fourth layer with 256 filters for deeper feature extraction. - **Filter Sizes**: Increased filters to capture more complex features. - **Frames Considered**: Using all frames for complete temporal information. - **Learning Rate**: Increased to 0.001 to aid convergence. - **Image Size**: Maintained at 128x128 for consistency. Model is clearly overfitting
3	4 Conv3D layers with filters [32, 64, 128, 256]	- **Image Size**: 128x128 - **Frames**: 30 - **Batch Size**: 32 - **Epochs**: 50 - **Learning Rate**: 0.001 - **Optimizer**: Adam - **Dropout**: 0.2 after Dense layers	Training accuracy: 0.69 Validation Accuracy: 0.58	- **Dropout Added**: Dropout layers (0.2) after Dense layers to prevent overfitting. - **Reason**: Helps in model generalization by reducing reliance on specific neurons. - **Other Parameters**: Same as Model 2 to compare the effect of Dropout.
4	4 Conv3D layers with filters [64, 128, 256, 512]	- **Image Size**: 112x112 - **Frames**: 15 (Alternate frames) - **Batch Size**: 16 - **Epochs**: 30 - **Learning Rate**: 0.0001 - **Optimizer**: Adam - **Dropout**: 0.5 after Dense layers	Training accuracy: 0.60 Validation Accuracy: 0.49	- **BatchNormalization Added**: Improves training speed and stability. - **Increased Filters**: Higher filters (up to 512) for complex feature learning. - **Image Size Reduced**: To 112x112 to reduce computational load. - **Dropout Increased**: 0.5 to further prevent overfitting. - **MaxPooling Adjustment**: Pooling only spatial dimensions to retain temporal

		- **BatchNormalization**: After each Conv3D layer		features. Increase in filters does not give a better result
5	4 Conv3D layers with filters [32, 64, 128, 256]	- **Image Size**: 128x128 - **Frames**: 30 - **Batch Size**: 32 - **Epochs**: 50 - **Learning Rate**: 0.0001 - **Optimizer**: Adam - **Dropout**: 0.4 after Dense layers - **BatchNormalization**: After Conv3D and Dense layers - **L2 Regularization**: 0.0005 on Conv3D layers	Training accuracy: 0.80 Validation Accuracy: 0.84	- **L2 Regularization Added**: Penalizes large weights to prevent overfitting. - **BatchNormalization**: Stabilizes and accelerates training. - **Dropout**: Increased to 0.4 for better regularization. - **EarlyStopping**: Added to prevent over-training. - **Filter Regularization**: Encourages smaller weights for generalization.
6	4 Conv3D layers with filters [32, 64, 128, 256]	**Image Size**: 128x128 - **Frames**: 30 - **Batch Size**: 32 - **Epochs**: 50 - **Learning Rate**: 0.0001 - **Optimizer**: Adam - **Dropout**: 0.5 after Dense layers - **BatchNormalization**: After Conv3D and Dense layers - **L2 Regularization**: 0.0005 on Conv3D layers - **EarlyStopping Patience**: Increased to allow more training time	Training accuracy: 0.65 Validation Accuracy: 0.60	**Increased Dropout**: To 0.5 for stronger overfitting prevention. - **EarlyStopping Adjusted**: More patience to let the model train longer. - **Other Parameters**: Same as Model 5 to assess impact of adjustments. Training stopped after 38 epoch due to no improvement in validation loss
Experiments with CNN-RNN models				
7	Basic CNN-GRU Model with filters [32,	- **Image Size**: 128x128 - **Frames**: 15 (Alternate frames)	Training Accuracy: 0.62 Validation	- **Conv2D Layers**: Three layers capture spatial features progressively for all frames. - **Filter Sizes**: Increasing filters (32→64→128) and GRU (16) gate for

	64, 128] with GRU[16] gate	- **Batch Size**: 32 - **Epochs**: 20 - **Learning Rate**: 0.001 - **Optimizer**: Adam	Accuracy: 0.48	learning complex patterns. - **Image Size**: 128x128 balances detail and computational load. - **Frames Considered**: Alternate frames reduce redundancy and computation. - **Learning Rate**: Low rate for gradual convergence. There is high bias in the model evident thru' low accuracies. We need to increase the Conv2D layers for better feature extraction and GRU units for higher temporal accuracy. Also reduce image size to accelerate the training time and check the effect on overfitting.
8	As above with GRU[64]	- **Image Size**: 64x64 - **Frames**: 15 (Alternate frames) - **Batch Size**: 32 - **Epochs**: 25 - **Learning Rate**: 0.001 - **Optimizer**: Adam - **Trainable Parameters:** 2,215,749	Training Accuracy: 1.0 Validation Accuracy: 0.72	- **Image Size**: Reduced to 64x64 to reduce trainable parameters. - **GRU units**: Increased to 64 to improve temporal accuracy. Model is clearly overfitting. Try Drop-out to reduce overfitting and BatchNormalisation to improve training speed
9	CNN-GRU Model with Batch Normalisation with GRU[128]	- **Image Size**: 64x64 - **Frames**: 15 (Alternate frames) - **Batch Size**: 32 - **Epochs**: 40 - **Learning Rate**: 0.001 - **Optimizer**: Adam - **Dropout**: 0.5 after Dense layers - **Batch Normalization**: After Conv2D layers - **Trainable Parameters:** 193,413	Training Accuracy: 0.92 Validation Accuracy: 0.8	- **Dropout Added**: Dropout layers (0.5) added after GRU[128] layer to prevent overfitting. - **BatchNormalization Added**: Improves training speed and stability. - **Flatten()** was replaced with GlobalAveragePooling2D() layer to reduce spatial dimensions Overfitting slightly improved with increased validation accuracy. Trainable parameters reduced drastically with faster training speed. Try pre-trained model to improve training accuracy further
10	CNN-GRU Transfer Model with GRU[128]	- **Image Size**: 64x64 - **Frames**: 15 (Alternate frames) - **Batch Size**: 32 - **Epochs**: 20 - **Learning Rate**:	Training Accuracy: 1.0 Validation Accuracy:	- **Pre-Trained Model Added**: MobileNetV2 with Weights='imagenet' excluding top and freezing CNN layers added to improve accuracy and training speed Training accuracy reached maximum (1.0),

		0.001 - **Optimizer**: Adam - **Dropout**: None - **Batch Normalization**: None - **Pre-Trained Model**: MobileNetV2 - **Trainable Parameters:** 542,085	0.8	but overfitting stayed with relatively lower val accuracy. Try LSTM model next to observe accuracy levels.
11	Basic CNN-LSTM Model with LSTM[128 units]	- **Image Size**: 64x64 - **Frames**: 15 (Alternate frames) - **Batch Size**: 32 - **Epochs**: 20 - **Learning Rate**: 0.001 - **Optimizer**: Adam - **Dropout**: None - **Batch Normalization**: None - **Pre-Trained Model**: None - **Trainable Parameters:** 2,388,421	Training Accuracy: 1.0 Validation Accuracy: 0.7	- **LSTM replaced GRU**: LSTM gate added to replace GRU to improve validation accuracy Training accuracy reached maximum (1.0), but overfitting occurred from initial epochs itself. Try LSTM with pre-trained CNN model next to observe accuracy levels.
12	CNN-LSTM Transfer Model with LSTM[128 units]	- **Image Size**: 64x64 - **Frames**: 15 (Alternate frames) - **Batch Size**: 32 - **Epochs**: 20 - **Learning Rate**: 0.001 - **Optimizer**: Adam - **Dropout**: None - **Batch Normalization**: None - **Pre-Trained Model**: ResNet50 - **Trainable Params:** 1,115,269	Training Accuracy: 0.58 Validation Accuracy: 0.47	- **Pre-Trained Model Added**: ResNet50 with Weights='imagenet' excluding top and freezing CNN layers added to improve accuracy and training speed There was severe underfitting with training accuracy reaching only a max of 0.58. ResNet50 didn't go well with LSTM gates. Check standard LSTM model with regularisers.
13	CNN-LSTM Model with Regularisation with LSTM[128 units]	- **Image Size**: 64x64 - **Frames**: 15 (Alternate frames) - **Batch Size**: 32 - **Epochs**: 40 - **Learning Rate**: 0.001	Training Accuracy: 0.98 Validation Accuracy: 0.76	- **Dropout Added**: Dropout layers (0.25) added after each layer to prevent overfitting. - **Flatten()** was brought back replacing GlobalAveragePooling2D() layer to avoid loosing important spatial information Underfitting issue was addressed. But the

		- **Optimizer**: Adam - **Dropout**: 0.25 after each CNN layer - **Batch Normalization**: None - **Kernel_regularizer**:l2(0.001) - **Trainable Parameters:** 2,388,421		model still overfits at higher accuracies. So far, it was observed that pre-trained model MobileNetV2 yielded good training accuracy, but validation accuracy wasn't good with basic RNN (GRU) model. A basic LSTM model provided better training accuracy, but has high training time with overfitting at higher accuracy level. We would try now with higher complexities of Conv2D model with varying image size, batch size and regularization parameters to improve accuracy and reduce overfitting.
14	Multi-Layer CNN-GRU Model with Bidirectional GRU[32 units]	- **Image Size**: 64x64 - **Frames**: 15 (Alternate frames) - **Batch Size**: 32 - **Epochs**: 30 - **Learning Rate**: 0.001 - **Optimizer**: Adam - **Dropout**: 0.25 after each CNN layer - **Batch Normalization**: Done - **Kernel_regularizer**:l2(0.001) - **Trainable Parameters:** 642,501	Training Accuracy: 1.0 Validation Accuracy: 0.81	**Additional Conv3D Layer**: Fourth and fifth layers with 128 filters added for deeper feature extraction. Additional Dense Layer (256 units) added at the end of CNN layers. **Bidirectional GRU**: To preserve information and learn from **both past and future** videos. The model performance improved well in this configuration with Val accuracy reaching 0.81. But the model still overfits slightly at higher accuracies. Try with higher image resolution, higher frame count and batch size to improve accuracy further.
15	Multi-Layer CNN-GRU Model with Bidirectional GRU[32 units]	- **Image Size**: 64x64 - **Frames**: 10/ 15 / 30 (all frames) - **Batch Size**: 32 / 64 / 128 - **Epochs**: 30 - **Learning Rate**: 0.001 - **Optimizer**: Adam - **Dropout**: 0.25 after each CNN layer - **Batch Normalization**: Done - **Kernel_regularizer**:l	Training Accuracy: Varied Validation Accuracy: Varied	**Increased Batch Size:** • At batch size 128, GPU couldn't load the batch due to memory shortage. • At batch size 64 with all 30 frames, GPU couldn't handle the load again. • At batch size 64 with 15 frames, GPU could handle the processing load; but there was drastic overfitting as validation accuracy couldn't exceed 0.25 with peak training accuracy at 0.7 • There was underfitting when frame counts per video was reduced to 10. **Hence batch size was brought back at 32**

		2(0.001) - **Trainable Parameters:** 642,501		**and frame count was frozen at 15 (every alternate frame)**. Try increasing GRU units to improve validation accuracy.
16	Multi-Layer CNN-GRU Model with Bidirectional GRU[128 units]	- **Image Size**: 64x64 - **Frames**: 15 (Alternate frames) - **Batch Size**: 32 - **Epochs**: 30 - **Learning Rate**: 0.001 - **Optimizer**: Adam - **Dropout**: 0.25 after each CNN layer - **Batch Normalization**: Done - **Kernel_regularizer**:l2(0.001) - **Trainable Parameters:** 884,229	Training Accuracy: 1.0 Validation Accuracy: 0.78	**GRU units increased to 128**: To improve validation accuracy. The model performance reduced slightly with Val accuracy coming down to 0.78 and the model still overfitting at higher accuracies. The trainable parameters also increased causing longer training time. Try with higher image resolution while keeping the GRU units at 32.
17	Multi-Layer CNN-GRU Model with Bidirectional GRU[32 units]	- **Image Size**: 128x128 - **Frames**: 15 (Alternate frames) - **Batch Size**: 32 - **Epochs**: 30 - **Learning Rate**: 0.001 - **Optimizer**: Adam - **Dropout**: 0.25 after each CNN layer - **Batch Normalization**: Done - **Kernel_regularizer**:l2(0.001) - **Trainable Parameters:** 1,035,717	Training Accuracy: 1.0 Validation Accuracy: 0.9	**Image resolution increased to 128**: To improve validation accuracy. The model performance improved well with Val accuracy improved to 0.90 with little gap between training and validation accuracies indicating less overfitting. Though the trainable parameters increased to 1M+, but this is still reasonable for a video analysis model. This is the best result we have got.

Selected Best Model:
The model used for experiment no. 17 is the best performing model that we selected as outcome from this project. This model yielded a training accuracy of 100% and validation accuracy of 90%. The trainable parameters are reasonable at 1M+ thus suitable for quick deployment on small devices like TV.