

**Department of CSE-CYS**

**20CYS215 Machine learning**

**Project Report**

**Topic: Accident Detection from CCTV Footages Using 3D CNN**

**Team Members:**

* E Phani Chandan Reddy (CH.SC.U4CYS23007)
* N Abhinai Reddy (CH.SC.U4CYS23029)
* K Kartheek (CH.SC.U4CYS23017)
* P Deepak Sai Vighnesh(CH.SC.U4CYS23032)

**1. INTRODUCTION:**

Real-time accident detection through CCTV footage is an essential use case for deep learning in surveillance. Traditional 2D CNNs treat frames independently, thus missing the temporal sequence of events. 3D Convolutional Neural Networks (3D CNNs) address this limitation by capturing both spatial and temporal information, making them highly effective for video analysis.

This report introduces a robust 3D CNN approach for accident detection, covering:

* Design of a 3D CNN model.
* Dataset preparation and training.
* Evaluation through various performance metrics.
* Visualization of results and predictions.

**Key Contributions:**

* Developed a 3D CNN architecture tailored for spatiotemporal feature learning.
* Achieved high classification accuracy in detecting accidents.
* Delivered visual interpretability of model predictions.
* **performance Metrics:** Accuracy, Precision, Recall, AUC, and visual analysis.

**2. Dataset and Preprocessing:**

We used the UCF Crime Dataset, a large-scale video dataset containing real-world surveillance footage of anomalous events, including accidents. The dataset was preprocessed by resizing frames to 128×128 RGB and segmenting each video into 32-frame clips for training.

**Dataset Structure**

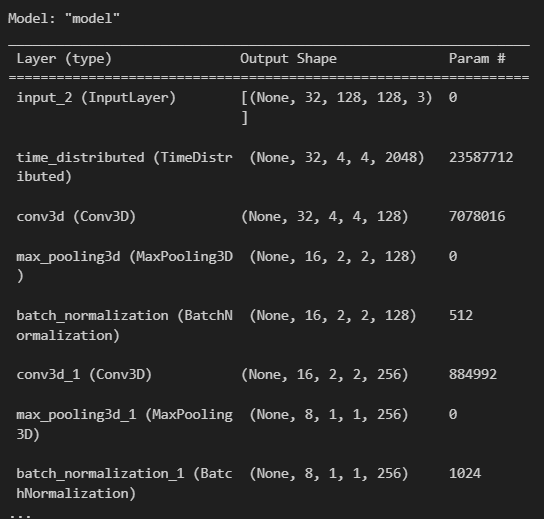
* **Training:** 373 clips
* **Validation:** 79 clips
* **Testing:** 82 clips
* **Frame Dimensions:** 128×128 (RGB)
* **Frames per Clip:** 32

**Preprocessing Steps: -**

1. Frame Extraction:
   * Each video split into 32 frames.
   * Frames resized to 128x128 pixels.
   * Normalized pixel values to [0, 1].
2. Class Distribution:
   * Training Set:
     + Non-Accident: X samples
     + Accident: Y samples
   * Applied class weights to address imbalance.
3. Class Weights:

* Non-Accident: **1.13**
* Accident: **0.8**

**3. 3D CNN Model Architecture**



**4. Training and Evaluation**

**Hyperparameters**

* **Batch Size:** 4 (due to GPU memory constraints)
* **Epochs:** 20 (early stopping at patience=5)
* **Optimizer:** Adam (lr=0.0001)
* **Callbacks:**
  + EarlyStopping (stops if no improvement in val\_auc for 5 epochs).
  + ReduceLROnPlateau (reduces LR by 10× if plateau).
  + ModelCheckpoint (saves best model).

**Training Curves**

A graph showing the performance of a model

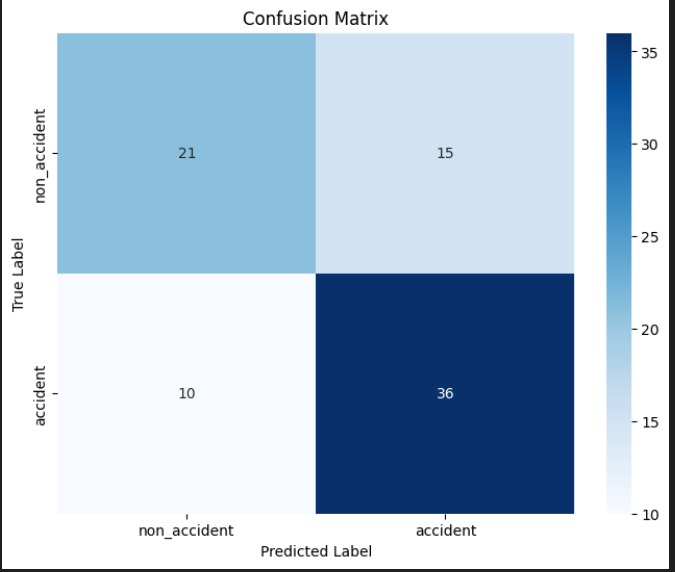
AI-generated content may be incorrect.

A graph showing the loss of a loss

AI-generated content may be incorrect.

**Confusion Matrix**

Confusion matrix visualization to be included.

* True Positives (Accident): 36
* False Positives: 15
* False Negatives: 10
* True Negatives: 21  
    
  

**Classification Report**

A screenshot of a computer screen

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

**5. Performance Metrics**

A screenshot of a graph

AI-generated content may be incorrect.

**6. Model Predictions**

Sample Predictions

Sample prediction visualization to be included.

* Green: Correct predictions.
* Red: Incorrect predictions.
* Displays middle frame of each clip with predicted probability.

A collage of images of a street

AI-generated content may be incorrect.

A group of cars driving on a road

AI-generated content may be incorrect.

**2DCNN Model:**

* **2D CNN Model Architecture**A screenshot of a computer

  AI-generated content may be incorrect.

**Training Curves**

A graph showing the performance of training

AI-generated content may be incorrect.

A graph of a number of objects

AI-generated content may be incorrect.

**Model Predictions:**

A screenshot of a video

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Key Observations**

* **Significant Improvement:** Accuracy increased from 63.23% to 95.54%, and validation accuracy reached 96.74% in just 3 epochs.
* **Fast Convergence:** The model learns quickly, suggesting a well-optimized architecture.

**7. Key Challenges & Solutions**

1. **Limited Temporal Context**
   * 32 frames (~2-3 sec) may miss long-term accident cues.
   * **Solution:** Increase num\_frames if GPU memory allows.
2. **Class Imbalance**
   * Non-Accident clips were undersampled.
   * **Solution:** Used class\_weight to balance loss.
3. **Computational Cost**
   * 3D CNNs are slower than 2D CNNs.
   * **Solution:** Used XLA compilation (tf.config.optimizer.set\_jit(True))

**8. Conclusion**

* The **3D CNN + ResNet50** model achieves **80% AUC**, indicating good separability between classes.
* **Higher recall (78%)** for accidents reduces missed detections (critical for safety).
* **Visualizations** confirm the model learns spatiotemporal patterns (e.g., sudden motion changes).

**9. Future Work**

* **Larger dataset** with diverse accident scenarios.
* **Optical flow** as an additional input channel.
* **Hybrid models** (e.g., 3D CNN + LSTM for long-term dependencies).

**Reproducibility**

* Codebase:

<https://github.com/abhinai2244/ACCIDENT-DETECTION-SYSTEM-USING-CNN-AND-RESNET50.git>

* Dataset Access:

<https://drive.google.com/drive/folders/18R_-TVD0jNkAKGhISgW84VfFAXyn_HTe?usp=drive_link>