2025

Final Project Report: Appendix

COS800027 – MACHINE LEARNING

THI NGAN HA DO	(103128918)
RATANAK PANHA DUONG	(104652159)
BAO MINH TRAN	(104763815)
MOHAMED SHARIQ USOOF	(104841889)
CHIN ZHAO JUN	(105210037)
HA NGO	(105256264)
AKSHAY CHAVAN	(105501919)

Task 3: Comparison

Pipeline Comparison

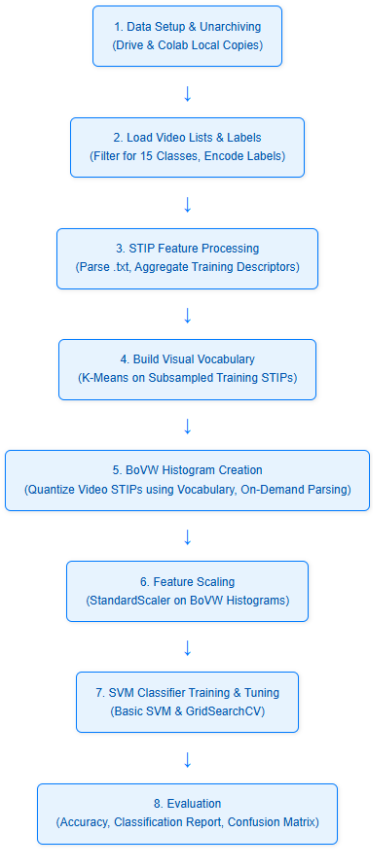

Figure 1. ML Pipeline Overview

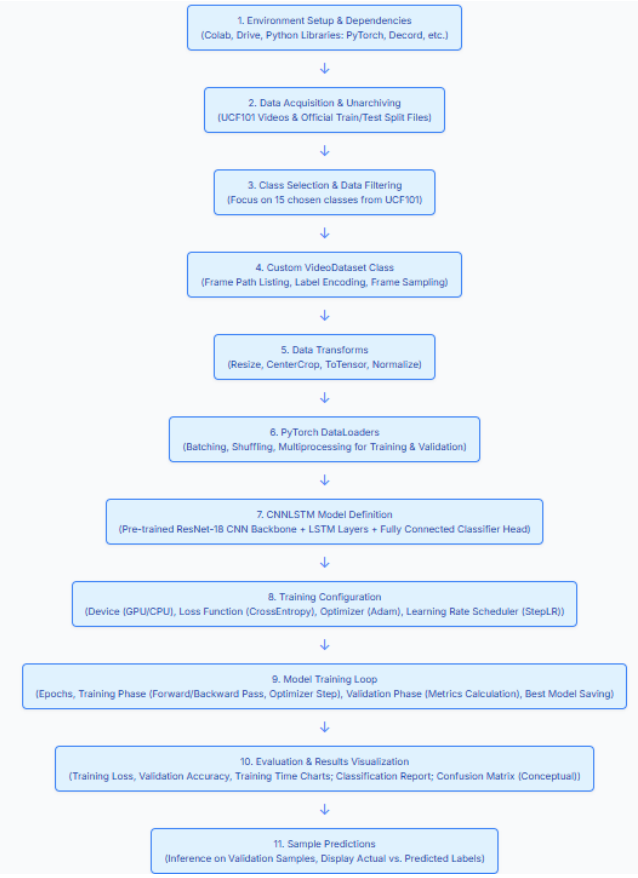

Figure 2. DL Pipeline Overview

The two classification pipelines represent different approaches to tackling the task of action recognition. The core distinction lies in their feature extraction and temporal modeling strategies.

The Machine Learning (ML) pipeline relies on hand-crafted features, starting with the extraction of Spatio-Temporal Interest Points (STIPs) from video frames and the computation of HOG (Histogram of Oriented Gradients) and HOF (Histogram of Optical Flow) descriptors, which requires manual input to design. These engineered features are quantised via K-Means clustering to form Bag of Visual Words (BoVW) histograms, serving as global video representations.

The Deep Learning (DL) pipeline uses learned features and an end-to-end approach, where a CNN backbone (ResNet-18) automatically extracts hierarchical spatial features from raw video frames, removing the need for manual feature engineering. Temporal dynamics are modelled using an LSTM layer that processes the CNN-extracted features to capture sequential context across frames. The CNNLSTM model is trained holistically for spatial and temporal components.

Performance Comparison

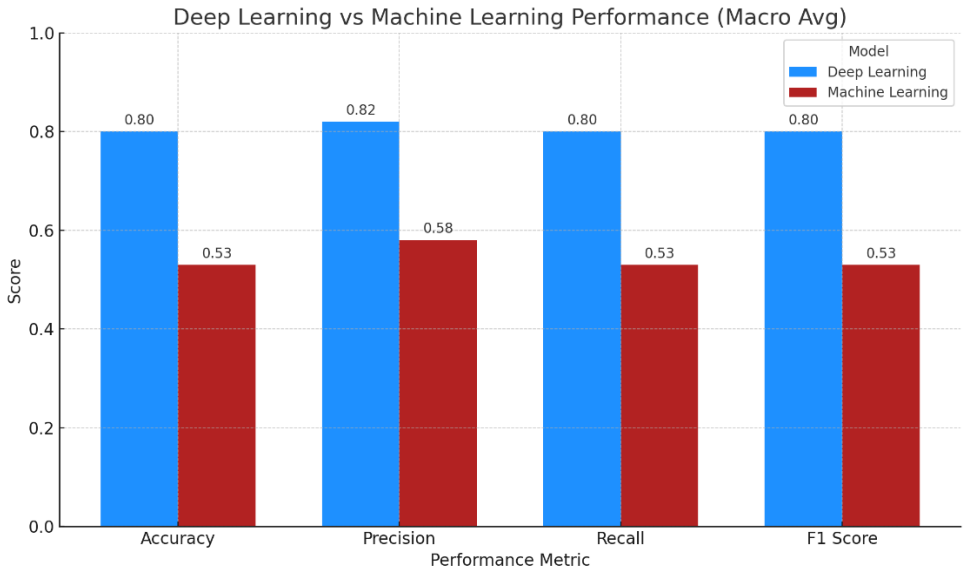

Figure 3. Model Performance Comparison

In terms of overall performance, the Deep Learning (DL) pipeline significantly outperforms the Machine Learning (ML) pipeline for video action classification. The DL model achieved an accuracy of 80%, and F1-scores of 0.80. In stark contrast, the ML model's best performance yielded an accuracy of 53%, with F1-scores around 53%.

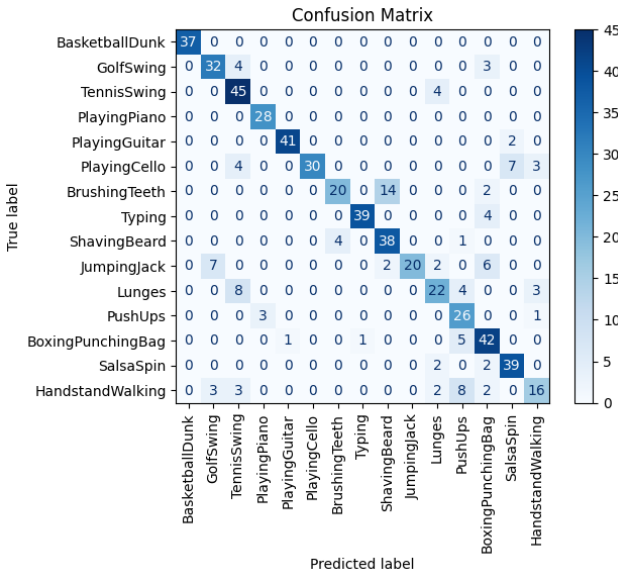

Figure 4. DL Confusion Matrix

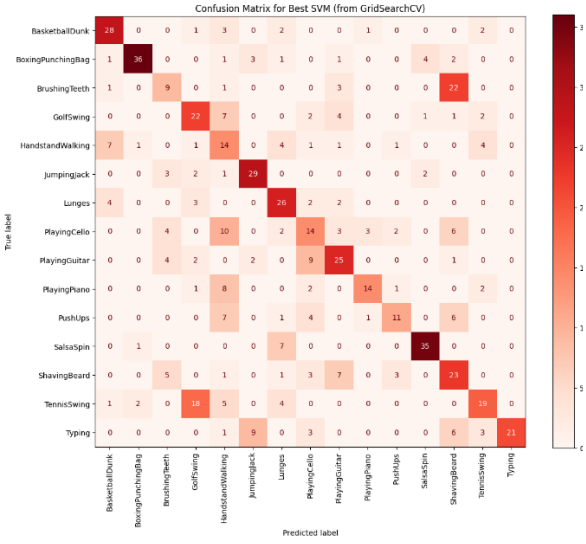

Figure 5. ML Confusion Matrix

When examining actions where each model struggles, distinct patterns emerge. The DL model tends to perform less optimally on actions with subtle or potentially ambiguous movements, such as 'Lunges' (F1-score: 0.42), 'BrushingTeeth' (F1-score: 0.61), and 'BoxingPunchingBag' (F1-score: 0.64). 'Lunges' might suffer due to varying execution styles or visual similarity to other body movements. 'BrushingTeeth' involves fine motor skills that could be challenging to capture consistently across diverse video contexts, and 'BoxingPunchingBag' might have internal

variations in punching techniques or camera angles that lead to confusion. Conversely, the ML model's lowest F1-scores are found in 'PlayingCello' (0.22), 'BrushingTeeth' (0.29), and 'HandstandWalking' (0.30). The poor performance on 'PlayingCello' and 'BrushingTeeth' likely stems from the inherent limitations of hand-crafted STIP features in capturing nuanced finger movements or subtle facial actions. 'HandstandWalking' is a highly dynamic and complex action, and the fixed nature of STIPs and the BoVW representation may not adequately encode the full spatio-temporal complexity and variations, leading to significant misclassifications.

Efficiency Performance

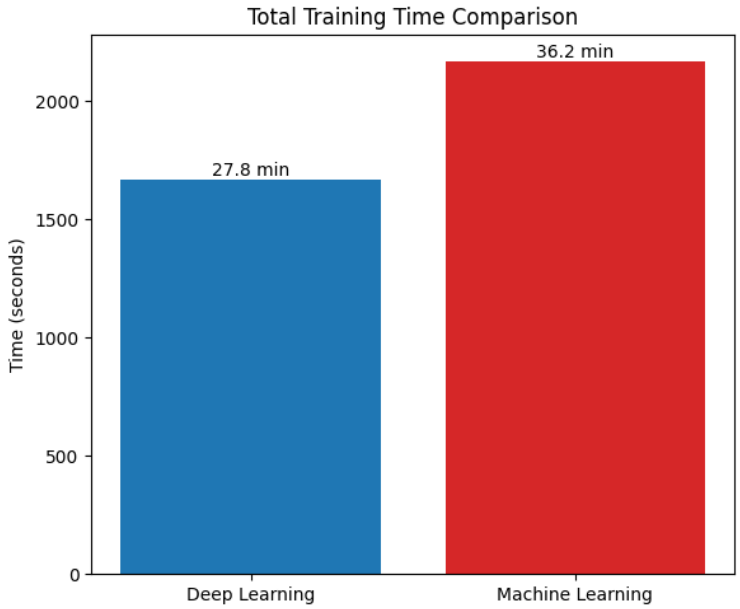

Figure 6. Total Training Time Comparison

Analyzing the training times, it is evident that the total runtime for ML, encompassing all preprocessing and feature engineering steps, was unexpectedly higher at approximately 36.2 minutes, despite the common expectation that traditional ML models are faster. This contrasts with the DL pipeline's total runtime of around 27.8 minutes. The pure model training time, however, tells a different story: the DL model's training (10 epochs at 166 seconds/epoch, totaling 27.8 minutes) was significantly longer than the ML model's SVM training, which was a mere 1.5 minutes (including GridSearchCV). This disparity arises because deep learning models involve iterative optimization over millions of parameters and complex non-linear operations across many layers, requiring extensive backpropagation through the entire network, which is computationally intensive even with GPU acceleration.

The longest component of the ML pipeline was the BoVW Histogram Generation, consuming approximately 27 minutes and 17 seconds. This is because it involves iterating through every video, parsing its STIP data (often from large text files), and then performing a quantization step (finding the nearest visual word) for potentially millions of individual STIP descriptors against a 500-word vocabulary, a process that is inherently sequential and I/O-bound. Therefore, while the DL model's iterative training phase is computationally demanding, its overall pipeline is

relatively quicker when considering all setup time, as it requires minimal manual preprocessing and feature engineering, with the model learning features directly from raw input.

Task 4: Ethics Considerations

4.1: Ethical Reporting and Dissemination

Model Limitations and Risks:

1. Deep learning Model (CNN + LSTM)

One of the significant limitations of this model is that it's black-boxed—i.e., although it can learn to do very well, it isn't too interpretable. It becomes difficult to understand what spatial-temporal patterns precisely the model is learning or how exactly it's arriving at a certain conclusion. Such lack of transparency could make debugging more difficult, and reduce the extent of user trust. Deep learning models are also data- and computationally-intensive, i.e., their performance is largely based on data quantity and quality. If the data set is imbalanced or limited, the model may overfit or generalize poorly to new actions.

2. Machine learning Model (STIP + BOVW + SVM)

This model, although simpler and more interpretable, presents several risks. It depends heavily on manually crafted features (STIP descriptors), potentially missing fine or complex motion clues in certain action classes. The Bag-of-Visual-Words (BoVW) approach discards temporal information completely, hence not working as well for actions with motion evolution. Additionally, it is sensitive to the way visual words are combined and selected, which can bring bias or overlook rare but informative patterns. Its lower complexity offers transparency at the cost of sacrificing some of the insightfulness of spatio-temporal video data.

Diversity and balance of data set:

The selected subset of the UCF101 dataset includes 15 action classes spanning various categories such as sports, musical performance, and human-object interaction, offering a reasonably diverse set of activities. However, some classes like “SalsaSpin” and “Typing” had noticeably fewer training examples compared to others like “PushUps” or “GolfSwing.” This class imbalance poses a risk of biased learning, where models may perform better on majority classes while underperforming on underrepresented ones. Such imbalance may also skew evaluation metrics and give a false sense of generalization.

Overfitting Concerns:

1. Deep Learning Model (CNN + LSTM):

Due to the high number of parameters in CNN + LSTM architectures, overfitting is a significant risk, especially given the limited number of videos per class. Despite applying techniques like dropout or data augmentation, the model may memorize dominant class patterns while failing to generalize to unseen variations or minority classes.

2. Machine Learning Model (STIP + BoVW + SVM):

While less prone to overfitting than deep models, this pipeline can still overfit if the visual vocabulary is too finely tuned to training examples or if SVM hyperparameters are not carefully regularized. The lack of temporal modeling further limits its generalization for dynamic actions, especially when training data is unevenly distributed across classes.

4.2: Transparency and Accountability

Dataset Selection and Justification

For this assignment, we were given the option to either find our own dataset or use the UCF101 dataset. We chose UCF101 as it is a well-established benchmark for human action recognition, offering over 13,000 labelled videos across 101 action categories. Finding another dataset of similar scale, diversity, and suitability for both machine learning and deep learning comparison tasks would have been relatively time-consuming. A subset of 15 diverse classes was selected in this dataset to balance task complexity with training efficiency.

Preprocessing and Model Choices:

When it comes to data loading, we addressed the challenge of managing high-gigabyte files by leveraging Google Colab, which allowed us to run intensive tasks in a cloud environment. For accessibility, we mounted Google Drive and optionally stored large datasets there, allowing us to load and store data during training and evaluation. This approach was intentionally flexible to accommodate group members who preferred running the code either locally or within the Colab environment.

Preprocessing involved extracting features from videos using either traditional descriptors (STIP for Machine Learning task) or deep learning pipelines (frame extraction for the CNN+LSTM model). For Machine Learning Model task, feature quantization via Bag-of-Visual-Words was applied. Model choices were guided by task suitability: STIP+BoVW+SVM offers interpretability and speed, while CNN+LSTM captures complex spatiotemporal dynamics.

Use of Pre-trained Models:

The deep learning model employed a pre-trained ResNet-18 as a spatial feature extractor. This choice was justified by ResNet's performance on datasets like ImageNet, which helped overcome limited training data by using low-level features. We ensured that the pre-trained ResNet-18 model was not trained on any UCF-101 or related action recognition datasets, thereby adhering to the assignment's requirement to avoid pre-training on similar data.

Reproducibility of Results:

All preprocessing steps, model configurations, and hyperparameters were documented and coded within a structured Colab pipeline, ensuring full reproducibility. Feature extraction outputs were saved in standard formats (e.g., .pkl), and model training scripts used fixed seeds where applicable.

Appendix

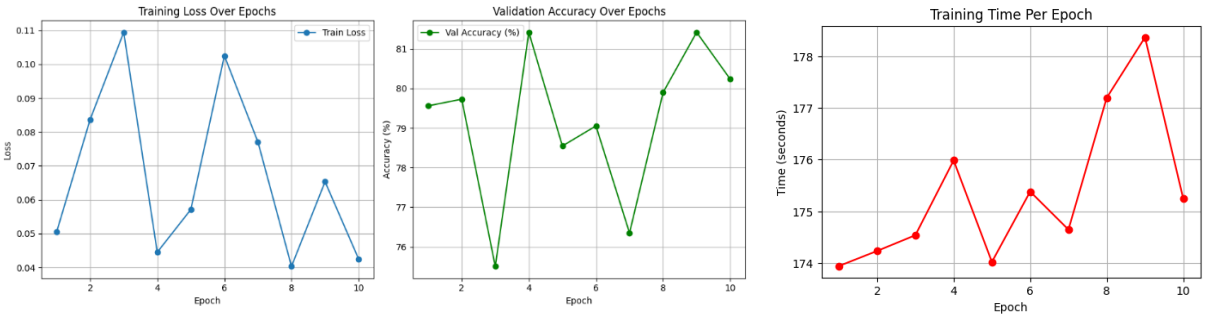

Figure 7. DL Training Progress