University of Milano – Bicocca
Master's Degree in Data Science
Couse: Foundations of Deep Learning
Academic Year 2023 - 2024

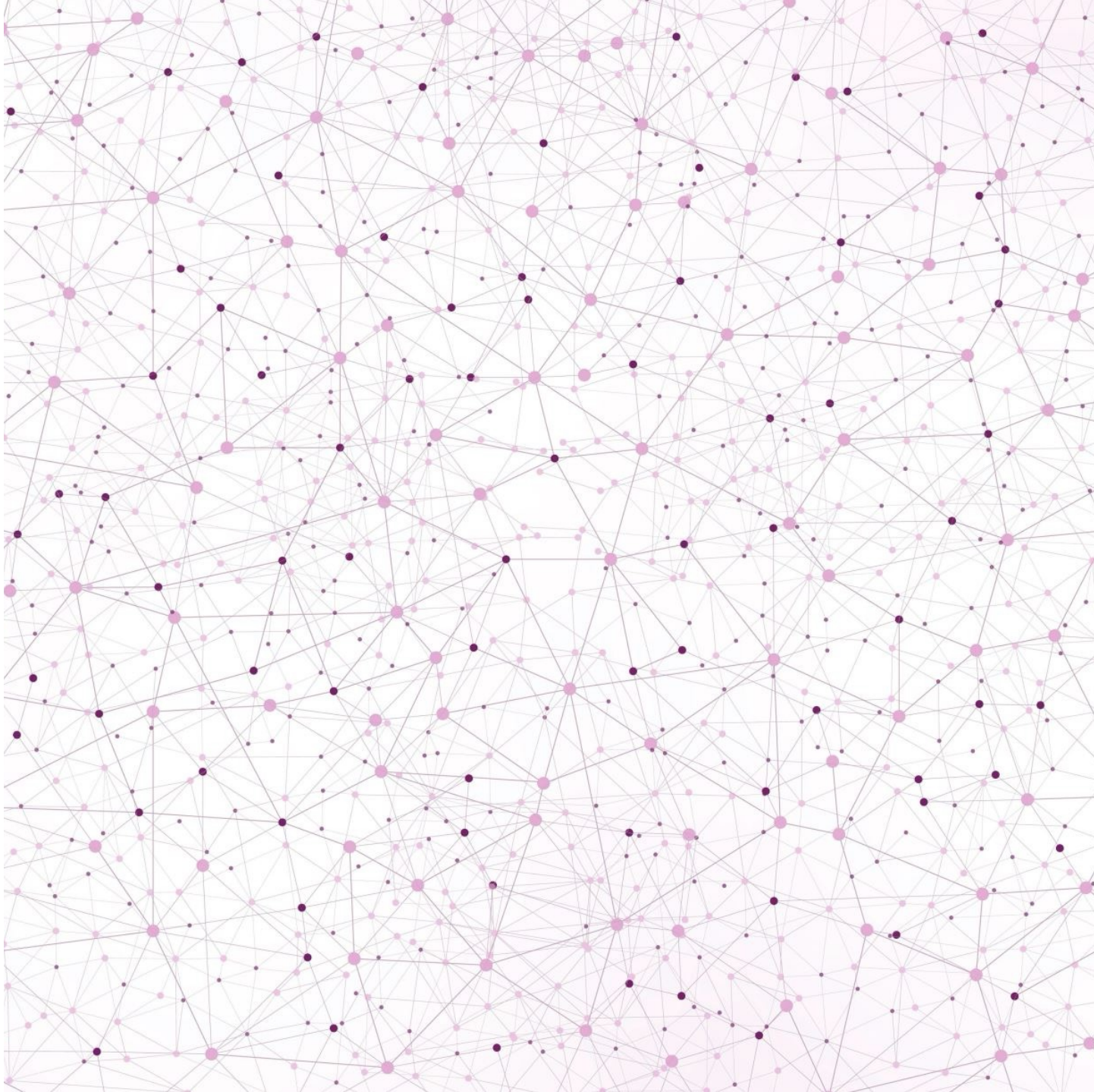
HUMAN ACTION RECOGNITION

VIDEO CLASSIFICATION ON HMDB51 DATASET

Project Members:

Giulia **Saresini**, mat. **864967**
g.saresini@campus.unimib.it

Sara **Nava**, mat **870885**
s.nava38@campus.unimib.it



From Image to Video Classification

"***A video is a sequence of images*** *(called frames) captured and eventually displayed at a given frequency. However, by stopping at a specific frame of the sequence, a single video frame, i.e. an image, is obtained.*"

Fig. 1: From the dataset HMDB51, HP_PRISONER_OF_AZKABAN_punch_u_cm_np1_ba_med_24.avi, class «punch».

From Image to Video Classification

"*A video is a sequence of images (called frames) captured and eventually displayed at a given frequency. However,* ***by stopping at a specific frame of the sequence****, a single video frame, i.e.* ***an image, is obtained***."

Fig. 2: From the dataset HMDB51, HP_PRISONER_OF_AZKABAN_punch_u_cm_np1_ba_med_24.avi, class «punch».

Dataset Presentation

HMDB collected from various sources, mostly from movies, and a small proportion from public databases such as the Prelinger archive, YouTube and Google videos.

The dataset contains **6849 clips** divided into **51 action categories**, each containing a minimum of 101 clips, that can be grouped in five types. However, owing to machine constraints, we've selected a single class, **body movements for human interaction**, including 7 actions: **fencing, hug, kick someone, kiss, punch, shake hands, sword fight**.

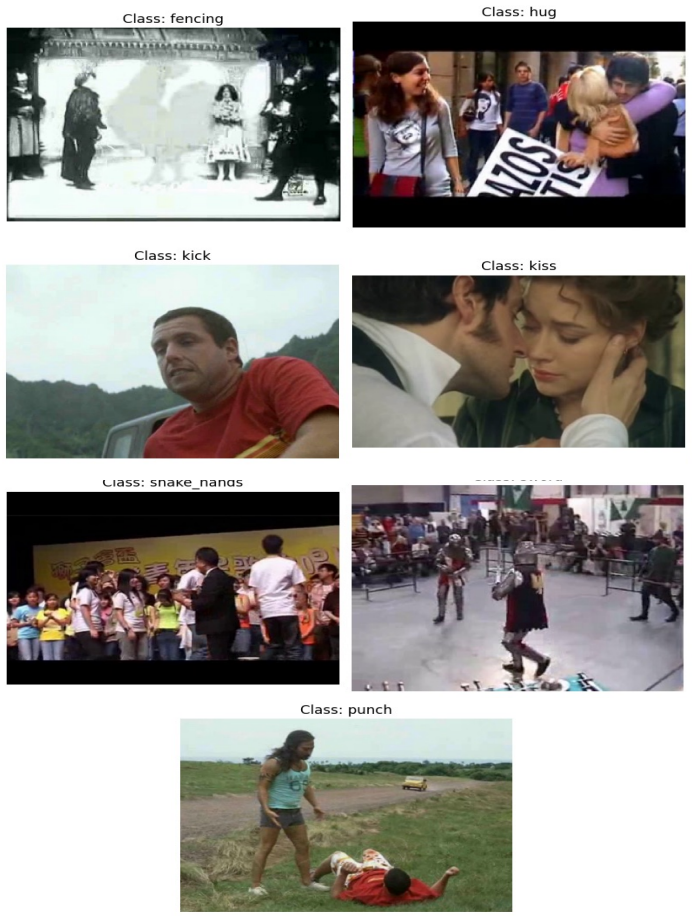


Fig. 3: From the dataset HMDB51, some random frames of the 7 selected classes.

Data Preparation

Given that the **video distribution** across each class **is evenly balanced** (between 100 and 150 videos per category), **we retained all original data** without discarding any information.

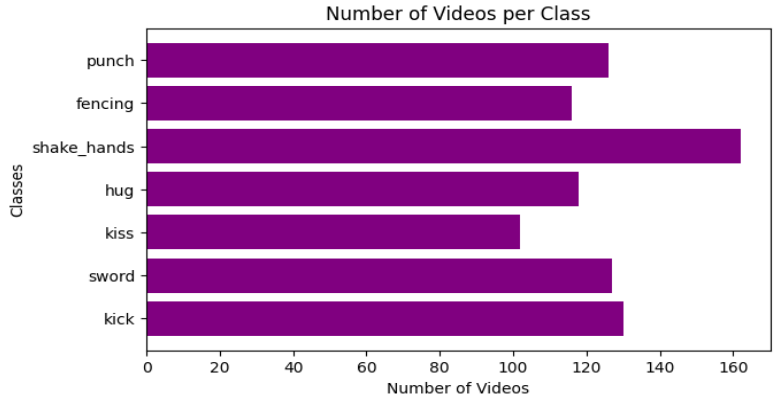


Fig. 4: Number of videos per classes

It was observed that **most videos contain over 200 frames**, while a **minority** have significantly **fewer** frames. For computational efficiency, we decided to **standardize** each **video** to **60 frames**. This approach involves two strategies:

- **duplicating existing frames if** a video has **fewer than 60** frames to meet the desired length
- otherwise, **it selects frames at regular intervals** to maintain consistency in frame count.

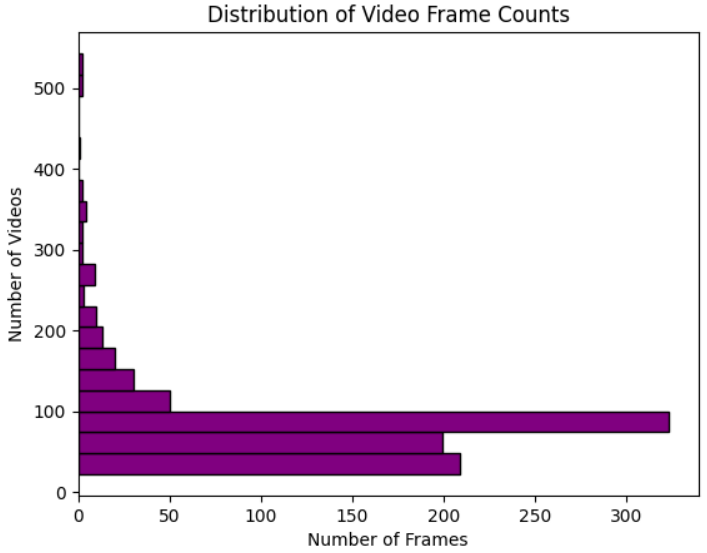


Fig. 5: Distribution of Video Frame Counts.

Data Preparation

Each frame was then **normalized** (divided by 255) so that each pixel assumed a value between 0 and 1, and subsequently **resized to a smaller dimension** (64x64) to remove redundant information, thereby reducing the computational resources required during training.

Fig. 6: Dimensionality Reduction.

3D Convolutional Neural Network

A **3D convolutional neural network** (3D CNN) is a type of neural network designed to **handle volumetric data**, such as video sequences.

Unlike 2D CNNs that use 2D filters on 2D images, 3D CNNs use **3D filters** that **slide over 3D volumes** (width, height, and depth), allowing the network to **capture spatio-temporal patterns**.

This enables 3D CNNs to analyze the temporal aspects of data, making them particularly **effective for** tasks involving **sequences of images over time**.

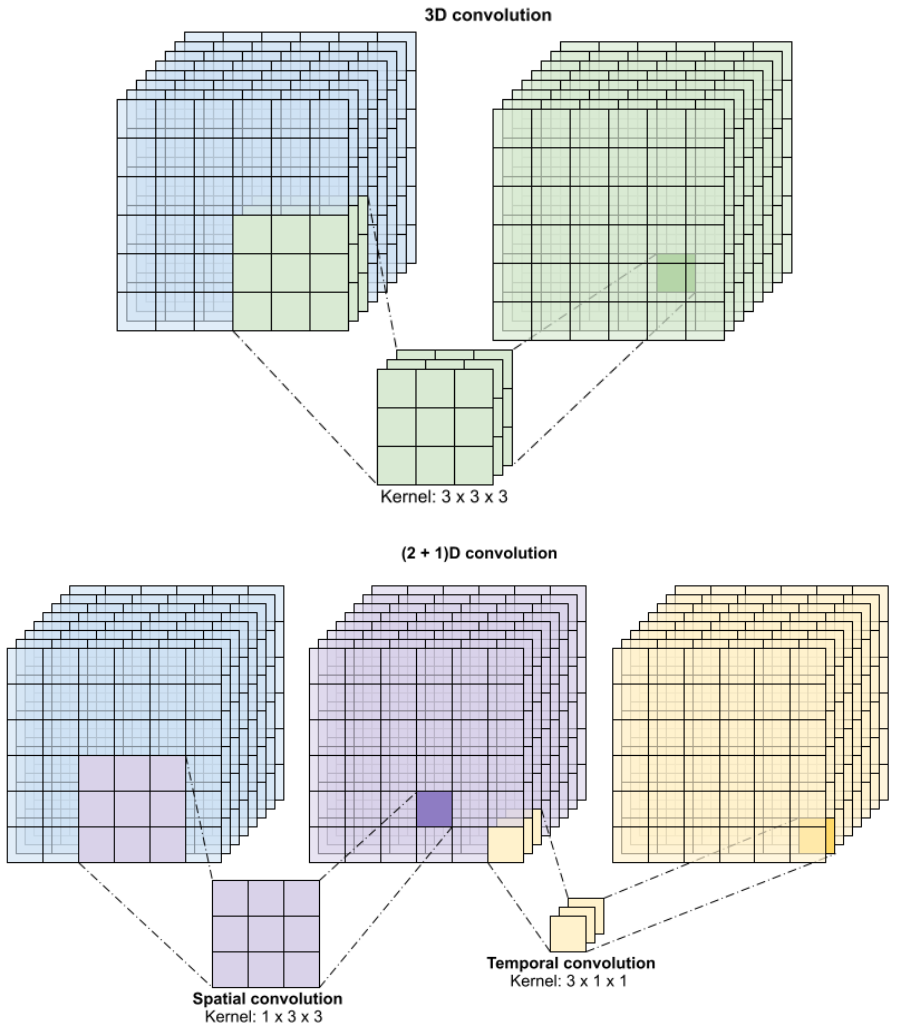


Fig. 7: 3D CNN.

3D Convolutional Neural Network

The network consists of the following layers:

- **Conv3D layers**, comprising two layers with kernel size (3, 3, 3) and 32 and 64 filters respectively.
- **MaxPooling3D layers**, including two layers with pool size (2, 2, 2).
- **Dropout layers**, consisting of three layers with a dropout rate of 50%.
- **Dense layers**, featuring one layer with 64 units and ReLU activation.
- **Output layer**, the final layer with softmax activation for multiclass classification.

Layer (type)	Output Shape	Param #
conv3d (Conv3D)	(None, 58, 62, 62, 32)	2,624
max_pooling3d (MaxPooling3D)	(None, 29, 31, 31, 32)	0
dropout (Dropout)	(None, 29, 31, 31, 32)	0
conv3d_1 (Conv3D)	(None, 27, 29, 29, 64)	55,360
max_pooling3d_1 (MaxPooling3D)	(None, 13, 14, 14, 64)	0
dropout_1 (Dropout)	(None, 13, 14, 14, 64)	0
flatten (Flatten)	(None, 163072)	0
dense (Dense)	(None, 64)	10,436,672
dropout_2 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 7)	455

Fig. 8: 3D CNN structure.

3D Convolutional Neural Network

The model does not achieve optimal performance as shown in both graphs. Despite the addition of L2 regularization and dropout, there is still **slight overfitting.**

Here are the achieved results:

- Best Training Accuracy: **50,15%**
- Best Validation Accuracy: **47,73%**
- Best Test Accuracy: **43,15%.**

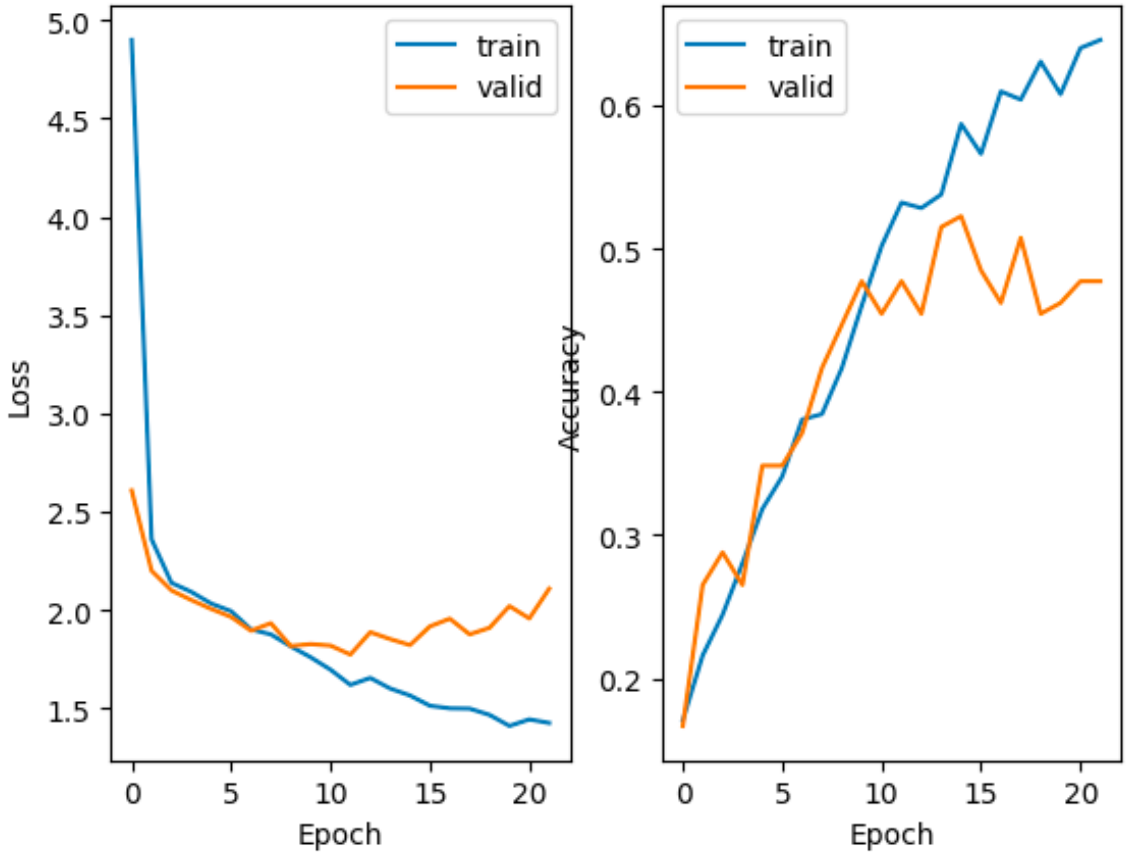


Fig. 9: Loss and Accuracy history during the training process.

Long-term Recurrent Convolutional Network (LRCN)

Long-term Recurrent Convolutional Network (LRCN) combines convolutional and recurrent layers to **handle sequences of data**, such as video streams.

Convolutional layers capture spatial features, while recurrent layers model temporal dependencies across sequences. This enables LRCNs to process variable-length.

LRCNs excel in tasks like activity recognition, image captioning, and video description by learning both spatial and temporal representations, offering advantages over traditional models that treat these aspects separately

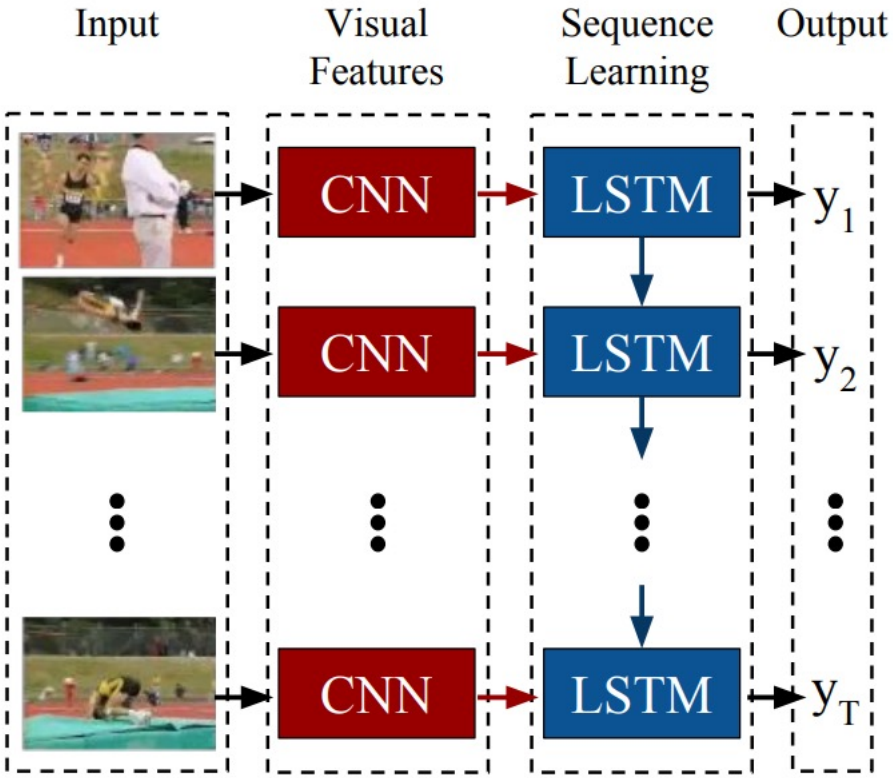


Fig. 10: LRCN structure.

Long-term Recurrent Convolutional Network (LRCN)

The network consists of the following layers:

- **TimeDistributed Wrapper:** Each CNN layer is wrapped in a TimeDistributed wrapper.
- **Convolutional Layers (Conv2D):** The network contains multiple convolutional layers with kernel sizes of (3, 3), 'same' padding, and ReLU activation function.
- **BatchNormalization:** applied to stabilize and accelerate the training process.
- **MaxPooling:** Max pooling layers with a pool size of (2, 2) reduce the spatial dimensionality of the feature maps.
- **Dropout:** Dropout layers with varying rates (0.3 and 0.4) are applied to prevent overfitting.
- **Flatten:** A Flatten layer converts the 2D feature maps into a 1D vector before passing them to the LSTM.
- **LSTM Layer:** A single LSTM layer with 64 units, a dropout rate of 0.5, and a recurrent dropout rate of 0.5 manages the temporal dependencies between frames.
- **Dense Layer:** The final dense layer with softmax activation classifies the video sequences into one of the 7 classes.

Long-term Recurrent Convolutional Network (LRCN)

Layer (type)	Output Shape	Param #
time_distributed_112 (TimeDistributed)	(None, 60, 64, 64, 16)	448
time_distributed_113 (TimeDistributed)	(None, 60, 64, 64, 16)	64
time_distributed_114 (TimeDistributed)	(None, 60, 32, 32, 16)	0
time_distributed_115 (TimeDistributed)	(None, 60, 32, 32, 16)	0
time_distributed_116 (TimeDistributed)	(None, 60, 32, 32, 32)	4,640
time_distributed_117 (TimeDistributed)	(None, 60, 32, 32, 32)	128
time_distributed_118 (TimeDistributed)	(None, 60, 16, 16, 32)	0
time_distributed_119 (TimeDistributed)	(None, 60, 16, 16, 32)	0
time_distributed_120 (TimeDistributed)	(None, 60, 16, 16, 32)	9,248
time_distributed_121 (TimeDistributed)	(None, 60, 16, 16, 32)	128
time_distributed_122 (TimeDistributed)	(None, 60, 8, 8, 32)	0
time_distributed_123 (TimeDistributed)	(None, 60, 8, 8, 32)	0
time_distributed_124 (TimeDistributed)	(None, 60, 8, 8, 64)	18,496
time_distributed_125 (TimeDistributed)	(None, 60, 8, 8, 64)	256
time_distributed_126 (TimeDistributed)	(None, 60, 4, 4, 64)	0
time_distributed_127 (TimeDistributed)	(None, 60, 4, 4, 64)	0
time_distributed_128 (TimeDistributed)	(None, 60, 1024)	0
lstm_6 (LSTM)	(None, 64)	278,784
dense_6 (Dense)	(None, 7)	455

Fig. 11: LRCN structure.

Long-term Recurrent Convolutional Network (LRCN)

Although slightly better than 3DCNN performances, this model does not achieve optimal performance as shown in both graphs. Despite the addition of L2 regularization and dropout, there is still **slight overfitting.**

Here are the achieved results:

- Best Training Accuracy: **86,71%**
- Best Validation Accuracy: **59,09%**
- Test Accuracy: **56,95%**

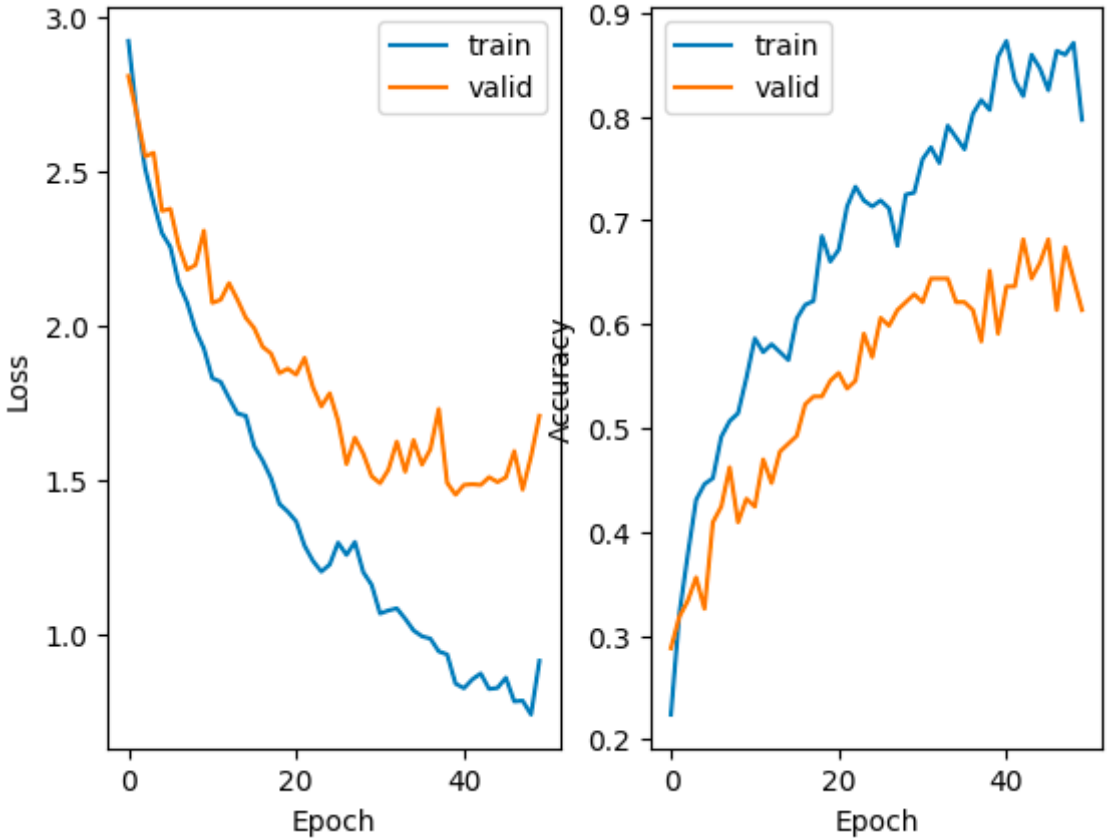


Fig. 12: Loss and Accuracy history during the training process.

Transfer Learning

Transfer Learning? Great Idea! But maybe with a **more powerful machine**...

We tried to leverage transfer learning by using a pre-trained model to extract image features and an LSTM to handle the temporal sequence. However, it **kept failing due to kernel breaks**. In particular:

1. **Feature Extraction**: Each frame of the video is processed by the ResNet50, which extracts the salient features of the image.
2. **Temporal Processing**: The features of all frames are analyzed by the LSTM, capturing the temporal dynamics and variations over time.
3. **Final Classification**: The temporal information processed by the LSTM is then used to determine the class of the video.

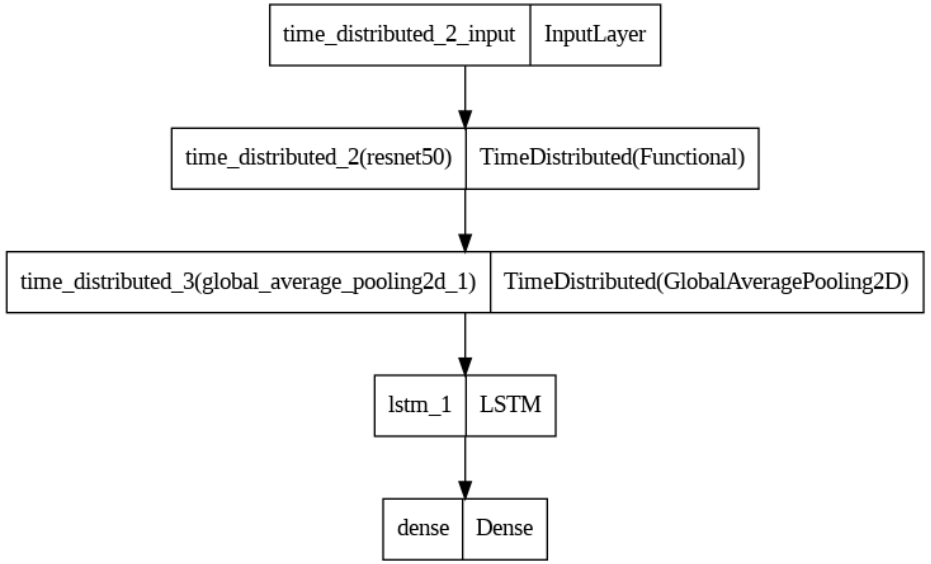


Fig. 13: Model Structure.

HMDB-51 Benchmark (Action Recognition)

We didn't have to look far to find models with better performance. For instance, VideoMAE V2-g achieved an impressive accuracy of 88.1%, and DEEP-HAL with ODF+SDF (I3D) reached 87.56%.

From the beginning, we knew that reaching the top spot wasn't an option for us. These results highlight a significant performance gap between our approach and the leading methods in action recognition on HMDB51, as showcased by the Papers With Code leaderboard.

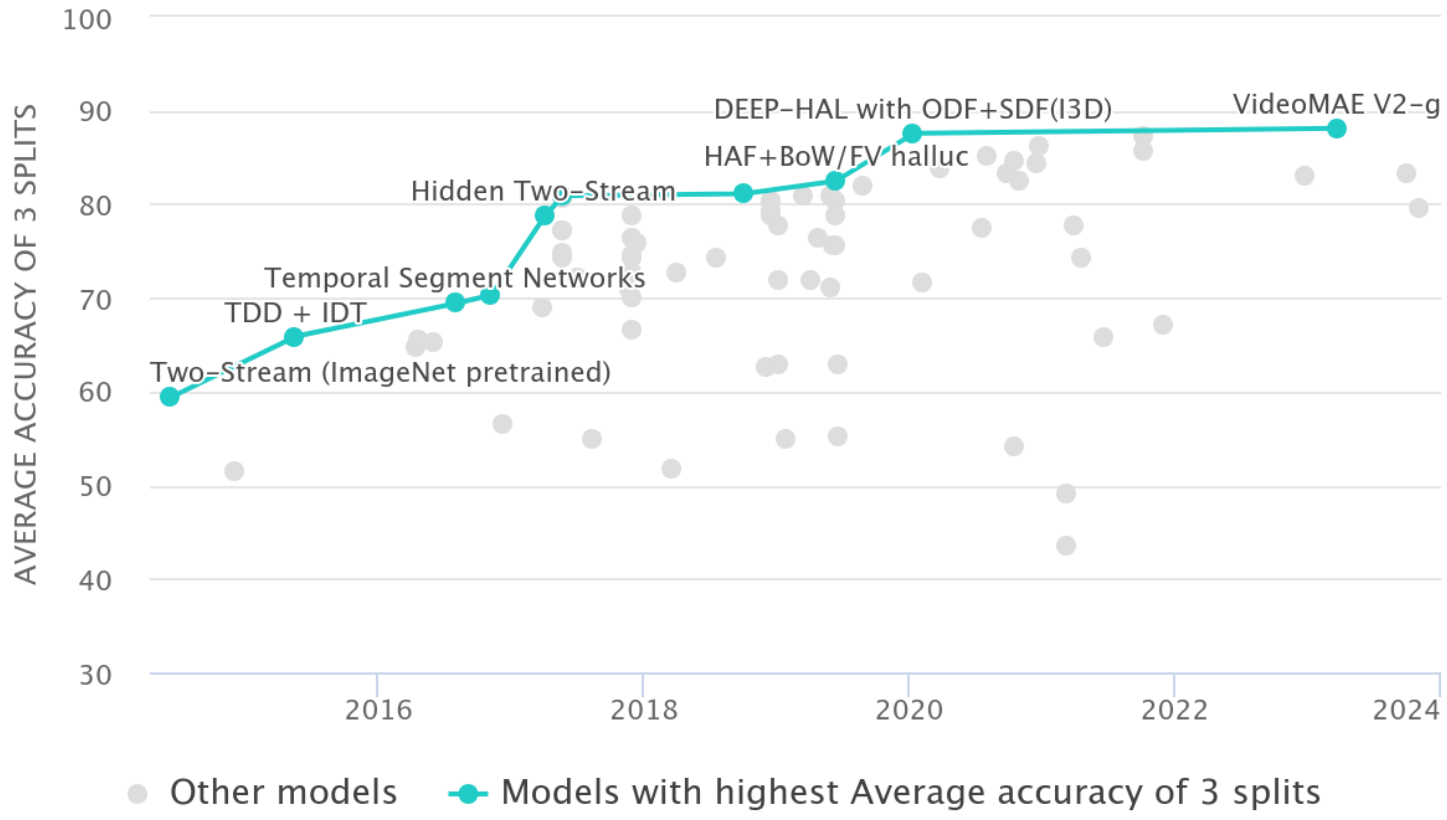


Fig. 14: Papers With Code leaderboard.

THANK YOU FOR THE ATTENTION!

Giulia Saresini & Sara Nava

References

HMDB-51 Dataset

Video Classification | Papers With Code

[1505.06250] Efficient Large Scale Video Classification (arxiv.org)

Introduction to Video Classification | by Connor Shorten | Towards Data Science

3D Convolutional Neural Network — A Guide for Engineers | Neural Concept

Understanding 1D and 3D Convolution Neural Network | Keras | by Shiva Verma | Towards Data Science

How does a 3D convolutional neural network differ from a 2D network in terms of dimensions and strides? - EITCA Academy

Video classification with a 3D convolutional neural network | TensorFlow Core

https://keras.io/api/applications/

Long-term Recurrent Convolutional Networks for Visual Recognition and Description

Time Distributed layer

Action Recognition on HMDB-51

Useful

Here it’s possible to find the trained models within this project:
https://drive.google.com/drive/folders/1F00GY0uog9f6-xQFowanqnlCZ1dg47Xr?usp=share_link