**HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**

SCHOOL OF INFORMATION COMMUNICATION TECHNOLOGY



**CAPSTONE PROJECT REPORT:**

**HAND-TRACKING COMPUTER CONTROL**

**Supervised by:**

Ph. D Tran Nguyen Ngoc

**Group member:**

|  |  |  |
| --- | --- | --- |
| Name | Roles | ID |
| Bui Khac Chien | Skeleton Extraction, MSSTNET | 20220059 |
| Trinh Hoang Anh | STMEM + Resnet3D | 20225470 |
| Tran Quang Minh | Resnet3D | 20225512 |
| Vo Ta Quang Nhat | Image Preprocessing, UI | 20225454 |
| Bach Nhat Minh | Shift GCN | 20225509 |

**TABLE OF CONTENTS**

[Abstract 3](#_Toc198207346)

[1. Introduction 4](#_Toc198207347)

[1.1. Problem Description 4](#_Toc198207348)

[1.2. Goals and Targets 5](#_Toc198207349)

[2. Dataset and Evaluation 6](#_Toc198207350)

[2.1. Dataset 6](#_Toc198207351)

[2.2. Data Preprocessing 9](#_Toc198207352)

[2.3. Evaluation Criteria 10](#_Toc198207353)

[3. Methodology 11](#_Toc198207354)

[3.1. Framework 11](#_Toc198207355)

[3.2. Skeleton-guided RGB frame cropping 12](#_Toc198207356)

[3.3. STMEM (Spatio-Temporal Motion Excitation Module) 13](#_Toc198207357)

[3.4. Resnet 14](#_Toc198207358)

[3.5. MSSTNet 14](#_Toc198207359)

[3.6. Training 14](#_Toc198207360)

[4. Result 14](#_Toc198207361)

[5. Application 16](#_Toc198207362)

[6. Discussion and Conclusion 16](#_Toc198207363)

[5.1. Discussion 16](#_Toc198207364)

[5.2. Conclusion 17](#_Toc198207365)

[7. References 18](#_Toc198207366)

**LIST OF FIGURES**

[Figure 1: Examples of videos from our dataset. Each image corresponds to a randomly sampled frame from a randomly sampled video. The image shows a large variance of the appearance of peoples, background scenes and occlusion in the videos. 8](#_Toc198240723)

[Figure 2: Class distribution of gesture class in Train and Validation sets 9](#_Toc198240724)

[Figure 3: The **hand skeleton** consists of **21 key landmarks** that represent important points on the hand. 10](#_Toc198240725)

[Figure 4: The framework of DSCNet 13](#_Toc198240726)

[Figure 5: Illustration of skeleton data 16](#_Toc198240727)

[Figure 6: MSST module consists of five convolutional branches. The entire network is composed of multiple MSST modules and average pooling modules. Pooling reduces the spatial size of the feature map and suppresses overfitting 18](#_Toc198240728)

[Figure 7: Input for MSSTNet model 19](#_Toc198240729)

[Figure 8: Training and Validation Accuracy/Loss on the Joint Stream Data 20](#_Toc198240730)

[Figure 9: Training and Validation Accuracy/Loss on the Joint-motion Stream Data 21](#_Toc198240731)

[Figure 10: Training and Validation Accuracy/Loss on the Bone Stream Data 21](#_Toc198240732)

[Figure 11: Training and Validation Accuracy/Loss on the Bone-motion Stream Data 21](#_Toc198240733)

**LIST OF TABLES**

[Table 1: 18 gesture labels along with their functions and corresponding keyboard shortcuts 9](#_Toc198240734)

[Table 2: Architecture of the MSSTNet 17](#_Toc198240735)

# **Abstract**

RGB-based human action recognition often underperforms in complex and dynamic environments due to its sensitivity to background clutter and lighting variations. However, the structural robustness of skeleton-based data offers a strong complementary modality. As a result, multi-modal approaches that combine RGB and skeleton inputs have gained significant traction in recent research. In this report, we investigate two representative methods: Shift Graph Convolutional Networks (Shift-GCN) and Dense-Sparse Complementary Network (DSCNet). DSCNet is designed to efficiently harness the complementary strengths of both modalities while maintaining low computational cost - achieving competitive performance compared to traditional GCN-based methods that may require over 100 GFLOPs. Our approach adopts dense sampling for RGB frames and sparse sampling for skeleton sequences, reflecting the data richness and efficiency needs of each modality. Using skeleton data as a spatial guide, we crop the key active regions from RGB frames to suppress background interference. To further reduce computational overhead, we introduce the Short-Term Motion Extraction Module (STMEM), which compresses temporally dense RGB sequences into a more compact form before processing. Concurrently, we design the Multi-Scale Spatial–Temporal Network (MSSTNet) to effectively capture motion dynamics from the sparse skeleton stream. We evaluate these methods on Jester Dataset across standard metrics, including Accuracy, Precision, Recall, and AUC, and provide a detailed analysis of their respective strengths and limitations. The report concludes by addressing several open challenges in multi-modal action recognition and benchmarking our methods against current state-of-the-art models.

**Keywords:** Action recognition, RGB-based, Skeleton-based

# **Introduction**

## Problem Description

Action recognition constitutes a fundamental task in computer vision with significant practical applications spanning intelligent surveillance, human - computer interaction, rehabilitation, and clinical assessment. In specialized domains such as sports and physical therapy, action recognition systems offer potential as expert alternatives, providing detailed performance feedback or quantitative evaluations of recovery progress.

Recent advancements have explored diverse data modalities - such as RGB video, skeleton coordinates, depth maps, and optical flow - to enhance recognition accuracy. RGB data offer rich visual cues like color and texture but are susceptible to variations in background and lighting. In contrast, skeleton data provide robust, abstract representations of human posture and movement, less affected by environmental noise. As such, integrating both RGB and skeleton modalities has emerged as a promising direction, enabling the exploitation of complementary features.

With the rise of deep learning, action recognition models have significantly progressed. For RGB inputs, techniques such as the two-stream network and Temporal Segment Networks (TSN) have improved temporal modeling, while 3D CNNs capture spatio-temporal features more naturally. Lightweight 2D-CNN alternatives enhanced with temporal modules have also gained attention due to their computational efficiency.

For skeleton-based recognition, approaches largely fall into three categories: RNNs for sequential modeling, CNNs that convert joint movements into structured images, and GCNs that naturally capture the relational dynamics among joints. GCNs, in particular, have become a dominant paradigm due to their alignment with the graph structure of human skeletons.

However, many existing approaches primarily emphasize the integration of network components to enable deep feature interactions, while often neglecting the complementary nature of RGB and skeleton modalities **at the data level.** Moreover, they tend to adopt established single-modality architectures without specific optimization for multimodal contexts, which can result in suboptimal recognition accuracy and increased computational cost.

In this report, we examine the effectiveness of two advanced multimodal models **Shift-GCN and DSCNet** which better leverage the complementary characteristics of RGB and skeleton data. By conducting comprehensive experiments on the 20Bn-Jester dataset and evaluating multiple performance metrics, including Accuracy, Precision, Recall, and AUC, we aim to assess how these models address the limitations of conventional single-modality or weakly integrated approaches. This analysis provides insights into their strengths and limitations in the context of multimodal action recognition.

* Section 2 presents an overview of the 20Bn-Jester dataset, details the skeleton feature extraction process, and outlines the evaluation metrics used.
* Section 3 describes the methodological frameworks and training procedures.
* Section 4 analyzes and compares the performance of the two selected models.
* Section 5 demonstrates the application of the model in a YouTube-based setting, including video preprocessing and related steps.
* Section 6 concludes the report by summarizing the current findings, highlighting state-of-the-art models, and discussing future research directions.

## Goals and Targets

1. **Primary Goals:** Comprehend the Fundamentals of Action Recognition

* *Establish an Effective Pipeline:* Identify an efficient workflow to recognize gestures using video or webcam input, ensuring high accuracy and real-time performance.
* *Understand Landmark Extraction:* Learn how to extract body landmarks and their connections, and understand how to store these values for use as model input or in preprocessing.
* *Analyze the Shift-GCN and DSCNet Frameworks:* Conduct an in-depth analysis of Shift-GCN and DSCNet, comparing their performance using standard evaluation metrics.

1. **Secondary Goals:**

* *Ensure System Robustness:* Handle gesture variations, background noise and lighting changes to improve generalization.

1. **Target Objectives:**

* *Develop a Hand-Tracking Computer Control Application: Create a user-friendly application that allows users to control software - initially focusing on YouTube - with low latency and high accuracy.*
* *Explore Cross-Field Applications*: Extend the application’s utility to other domains, including social media, robotics, and education, showcasing its versatility and real-world relevance.

# Dataset and Evaluation

### Dataset

For this project, we make use of the **20BN - Jester dataset** hosted on Kaggle, which is a subset of the complete JESTER dataset. This dataset comprises a large collection of densely annotated video clips that capture individuals performing predefined hand gestures in front of a laptop or webcam. It was built through crowdsourcing efforts involving a large number of contributors. The dataset is specifically designed to support the development and training of machine learning models for hand gesture recognition - including actions such as *sliding two fingers downward*, *swiping left or right*, and *drumming fingers*.

The dataset includes the following components:

* **Three main folders**:
* Train: contains 50,400 directories
* Validation: contains 7,047 directories
* Test: contains 6,981 directories

Each directory is named after a unique video\_id (e.g., “1”, “100”, etc.) and contains **37 frames** representing a single gesture video.



Figure 1: Examples of videos from our dataset. Each image corresponds to a randomly sampled frame from a randomly sampled video. The image shows a large variance of the appearance of peoples, background scenes and occlusion in the videos.

* **Three annotation files (CSV format)**:
* Train.csv, Validation.csv, and Test.csv: Each file maps video\_id to their corresponding gesture labels.

In total, the dataset spans **27 gesture classes**, with two additional categories - *“No gesture”* and *“Do other things”* - provided to help models differentiate between defined gestures and miscellaneous or unrecognized hand movements.

For the purpose of integrating gesture control with the YouTube application, we selected **18 specific gesture labels**. These are summarized in the following table, along with their intended functions and corresponding keyboard shortcuts.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Label** | **Functionality** | **Keyboard Shortcut** |
| 1 | Doing other things | x | x |
| 2 | No gesture | x | x |
| 3 | Rolling Hand Backward | Seek to previous chapter | Ctrl + <-- |
| 4 | Rolling Hand Forward | Seek to next chapter | Ctrl + --> |
| 5 | Shaking Hand | Stop detect gesture | x |
| 6 | Sliding Two Fingers Down | Decrease Volume | Arrow up |
| 7 | Sliding Two Fingers Left | Increase Volume | Arrow down |
| 8 | Sliding Two Fingers Right | Seek forward 5 secs | --> |
| 9 | Sliding Two Fingers Up | Seek backward 5 secs | <-- |
| 10 | Stop Sign | Toggle play/pause | k |
| 11 | Swiping Down | Toggle captions | c |
| 12 | Swiping Left | Switch to previous tab | Alt + <-- |
| 13 | Swiping Right | Switch to next tab | Alt + --> |
| 14 | Swiping Up | Next video | shift + n |
| 15 | Thumb Down | Toggle mute | m |
| 16 | Thumb Up | Toggle full screen | f |
| 17 | Turning Hand Clockwise | Increase playback rate | < |
| 18 | Turning Hand Counterclockwise | Decrease playback rate | > |

Table 1: 18 gesture labels along with their functions and corresponding keyboard shortcuts



Figure 2: Class distribution of gesture class in Train and Validation sets

### Data Preprocessing

In the DSCNet framework, we ensemble a skeleton-based action recognition network, MSSTNet, with an RGB-based model. First, sparse pose estimation is performed on the RGB frames. The extracted skeleton data is then used not only as input for the skeleton-based recognition but also to guide the cropping of relevant regions in the RGB frames, thereby enhancing spatial focus and improving recognition accuracy.

To extract human skeletal features, our team employed **Google’s Mediapipe framework**, a versatile and lightweight machine learning solution developed by Google AI Edge. Mediapipe offers cross-platform compatibility and real-time inference, making it ideal for applications on mobile, desktop, web, and IoT devices. Notably, it is open-source and free to use, which facilitates easy integration and deployment.

In our implementation, we use Mediapipe’s *Hand Detection* and *Human Pose Estimation* functionalities with RGB image inputs (assuming mirrored input from a front-facing camera). These functions return the (x, y, z) coordinates of body landmarks. For the Hand Detection module, we construct a skeletal model of the hand consisting of **21 key points per hand** (Figure 3).

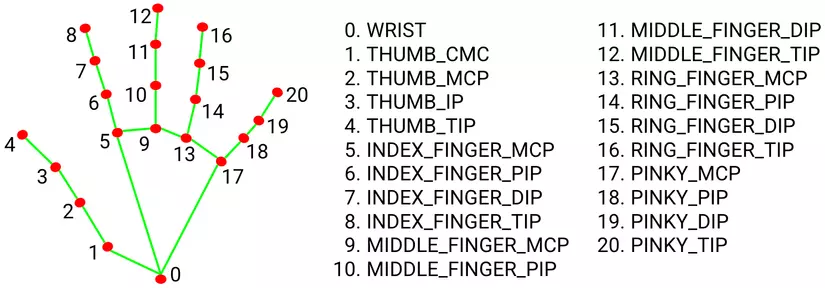


Figure 3: The **hand skeleton** consists of **21 key landmarks** that represent important points on the hand.

Regarding pose landmarks, since our work focuses specifically on hand-related gestures, we extract only **six upper-body landmarks**: *Left/Right Shoulder, Left/Right Elbow,* and *Left/Right Wrist.*

Each video consists of 37 frames. For each video, we extract features from the relevant landmarks and store the data in a tensor of shape (T, V, C) = (37, 48, 3), where:

* T is the number of frames,
* V is the number of joints (including 6 body pose landmarks and 21 for each left and right hand) and
* C represents the 3D coordinates (x, y, z). The same process is applied to both the training and validation datasets.

### Evaluation Criteria

To assess the performance of our action recognition models, we utilize four key evaluation metrics: **Accuracy**, **Precision**, **Recall**, and **AUC (Area Under the Curve)**. Each metric provides a different lens through which to measure model effectiveness.

1. **Accuracy**  
   Accuracy measures the overall correctness of the model. It is calculated as the ratio of correctly predicted actions to the total number of predictions.  
   This metric gives a general view of performance but may be misleading when classes are imbalanced.
2. **Precision**  
   Precision focuses on the quality of positive predictions. It is the proportion of true positive predictions among all positive predictions.  
   High precision means fewer false positives, which is critical in applications where incorrect positive results are costly.
3. **Recall**  
   Recall measures the model’s ability to detect all relevant instances. It is the ratio of true positives to all actual positives.  
   Recall (Sensitivity) is our **top priority**, as it measures the model’s ability to correctly identify true actions from users. In real-world applications, especially in gesture-based control systems, missing a user's intended action (false negative) can significantly degrade the user experience. Therefore, we aim for a high recall rate, ensuring that the system reliably captures all valid gestures, even in less-than-ideal conditions.
4. **AUC (Area Under the ROC Curve)**

AUC provides a comprehensive view of model performance across different thresholds. It represents the probability that the model ranks a randomly chosen positive instance higher than a randomly chosen negative one.

* An AUC of 1.0 indicates perfect separation.
* An AUC of 0.5 implies no discriminative ability.

# **Methodology**

#### Framework

In this report, we introduce DSCNet, an action recognition framework that integrates both RGB and skeleton-based modalities. The objective is to harness the complementary strengths of multimodal data while maintaining low computational overhead. The overall architecture of DSCNet is illustrated in Figure 4.

Given a video segmented into t = 37 frames, where each frame consists of channel (c), height (h), width (w) dimensions, we first apply a preprocessing step. The resulting data is structured in the format (batch size, T, V, C). This preprocessed skeleton data is then fed into MSSTNet for skeleton-based action recognition. Simultaneously, it is utilized to guide the cropping of relevant regions from the RGB frames.

The cropped RGB frames are divided into k = 6 segments, each containing 6 frames (excluding the final frame). These segments serve as input to the STMEM, which generates k motion maps. The resulting motion maps are then fed into an RGB-based action recognition backbone network. The final prediction is derived by fusing the classification outputs from both the skeleton and RGB streams, enabling a robust and comprehensive understanding of the performed actions.

RGB frames

Cropped RGB frames

Motions maps

A graph showing a bar graph

AI-generated content may be incorrect.

ResNet

STMEM

Cropping

A blue bar graph with a white background

AI-generated content may be incorrect.A blue bar graph with a white background

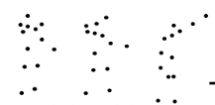
AI-generated content may be incorrect.A blue bar graph with a white background

AI-generated content may be incorrect.A blue bar graph with a white background

AI-generated content may be incorrect.

MSSTNET

Joint stream



Skeleton joints

Skeleton bones

Mediapipe

Joint-motion stream

Score Fusion

**+**

MSSTNET

Skeletons

A black stick figure with arms raised

AI-generated content may be incorrect.A group of people walking and walking

AI-generated content may be incorrect.

MSSTNET

Bone stream

MSSTNET

Bone-motion stream

(37, 48, 3)

A diagram of a person working on a computer

AI-generated content may be incorrect.

10 pixels

10 pixels

Figure 4: The framework of DSCNet

**Skeleton-guided Cropping**

#### Skeleton-guided RGB frame cropping

Due to the presence of numerous **action-irrelevant background elements** in RGB frames, directly extracting meaningful motion information becomes challenging. To address this, we employ a **skeleton-guided cropping strategy**.

In this approach, let denote the horizontal and vertical coordinates of the detected skeleton joints. Based on these coordinates, we determine the **active region** of the skeleton in the RGB frame, defined by the intervals [*min*(*X*), *max*(*X*)] and [*min*(*Y*), *max*(*Y*)] as illustrated by the **blue bounding box** in the lower-left corner of Figure 4. These define the width and height of the region where most of the motion occurs:

To ensure that the cropped region captures the full human body and allows for **data augmentation**, we apply padding: both the shorter side and the longer side of the bounding box are extended by 10 pixels. Furthermore, to **prevent overflow beyond the image boundaries**, the resulting box is clipped to remain within the original frame dimensions. The final cropping intervals for both horizontal and vertical axes can be expressed as:

[*max*(*min*(*X*) – 10, 1), *min*(*max*(*X*) + 10, *w*)]

and

[*max*(*min*(*Y*) – 10, 1), *min*(*max*(*Y*) + 10, *h*)]

Where the *w* and *h* are the width and height of the original RGB frame. The **red bounding box** in the bottom left of Figure 4 indicates this final cropping region. This region is then consistently applied to all sampled RGB frames, ensuring uniformity in background context.

This **skeleton-based cropping technique** effectively focuses the model’s attention on the regions of actual movement, thereby enhancing the performance of the STMEM module in action recognition tasks.

#### STMEM (Spatio-Temporal Motion Excitation Module)

Instead of directly feeding densely sampled RGB frames into the backbone network which incurs high computational costs, we adopt the Spatio - Temporal Motion Excitation Module (STMEM) to extract motion information segmentally. This approach compresses dense RGB frame sequences into a sparse representation, substantially reducing the data volume fed into the backbone. Within this module, convolutional neural networks are employed to capture motion cues over **short temporal intervals**. Notably, STMEM can be seamlessly integrated with backbone networks to form an end-to-end trainable action recognition framework, allowing its parameters to be optimized jointly for improved adaptability across diverse actions. The detailed architecture is illustrated in the bottom right of Figure 4. For *t* cropped frames in a video clip, we first calculate the frame differences:

A convolution layer is then used to extract the spatial-temporal features (*Fst*) of RGB frames and

where Conv3 denotes the convolution layer with the kernel size of 3 × 3.

Concatenating RGB frames retains sufficient spatial features compared to using alone. Then, to enhance crucial motion information, we use to generate a time attention mask through a maxpooling layer, a convolution layer, and a sigmoid function. The motion map is obtained by multiplying the mask and *Fst* in element levels. The entire process can be described as:

where maxpool denotes the maxpooling operation on the time dimension, denotes the sigmoid function, ⊙ represents element-wise multiplication.

#### Resnet

#### MSSTNet (Multi-scale spatial-temporal convolutional neural network)

We adopt MSSTNet as the baseline network for the skeleton modality. MSSTNet comprises seven MSST modules and two pooling modules, utilizing one-dimensional convolutions at multiple scales to effectively model the temporal and spatial dynamics of skeletal data. Originally designed for dense skeleton sequences, MSSTNet operates on inputs with a temporal length of 200 frames. However, in our implementation, we use sparser skeleton data with a temporal length of 37 frames.

Within MSST modules, stride settings are applied using convolution kernels of sizes 3 × 3, 5 × 1, 7 × 1, and 11 × 1, alongside an initial 1 × 1 convolution to manage the temporal resolution. All other convolutional layers maintain a stride of 1. The spatial resolution is primarily adjusted through the pooling modules and the stride of temporal convolutions, although for all MSST modules except the fifth, the stride is reduced to 1 to preserve temporal fidelity.

The skeleton data can be interpreted as a three-dimensional tensor of size H × W × C, where the height (H) is analogous to the temporal dimension (i.e., the number of frames, T), the width (W) represents spatial elements such as joints or bones (V), and the three channels (C) correspond to the spatial coordinates (x, y, z), similar to the RGB channels in an image (Figure 5).

*x*

*y*

*z*

*T*

*V*

Figure 5: Illustration of skeleton data

The architectural configuration of MSST modules can be summarized as follows:

* Input is fed into 5 parallel branches.
* Each branch begins with a 1×1 convolution to reduce dimensions.
* Branches have different convolutional patterns:
* Branch 1: Simple 1×1 convolution (preserves spatial information).
* Branch 2: 1×1 followed by separate 3×1 and 1×3 convolutions (small receptive field).
* Branch 3: 1×1 followed by separate 5×1 and 1×5 convolutions (medium receptive field).
* Branch 4: 1×1 followed by separate 7×1 and 1×7 convolutions (larger receptive field).
* Branch 5: 1×1 followed by separate 11×1 and 1×11 convolutions (largest receptive field).
* All branches are concatenated at the end to produce output feature maps.
* Each convolution is typically followed by batch normalization and ReLU activation.

The output of each module is the input of the next module. The detailed layout of MSSTNET is shown in Table 2 and Figure 6. MSSTNET is capable of learning motion patterns over short, medium, and long temporal intervals, and can distinguish between similar actions that differ in their motion details.

Table 2: Architecture of the MSSTNet

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Module | Stride | 1 x 1 | 3 x 3 | 5 x 1  1 x 5 | 7 x 1  1 x 7 | 11 x 1  1 x 11 | Input size |
| MSST module 1 | (1, 1) | 44 | 60 | 60 | 60 | 60 | (T, V, 3) |
| MSST module 2 | (1, 1) | 48 | 80 | 80 | 80 | 80 | (T, V, 284) |
| MSST module 3 | (1, 1) | 56 | 120 | 120 | 120 | 120 | (T, V, 288) |
| Avg pool | (1, 2) |  |  |  |  |  | (T, V, 536) |
| MSST module 4 | (1, 1) | 160 | 160 | 160 | 160 | 160 | (T, V/2, 536) |
| MSST module 5 | (2, 1) | 72 | 200 | 200 | 200 | 200 | (T, V/2, 800) |
| Avg pool | (1, 2) |  |  |  |  |  | (T/2, V/2, 872) |
| MSST module 6 | (1, 1) | 240 | 240 | 240 | 240 | 240 | (T/2, V/4, 872) |
| MSST module 7 | (1, 1) | 320 | 320 | 320 | 320 | 320 | (T/2, V/4, 1200) |
| Global Avg pool | (1, 1) |  |  |  |  |  | (T/2, V/4, 1600) |
| Full Connection |  |  |  |  |  |  | (1, 1, 1600) |

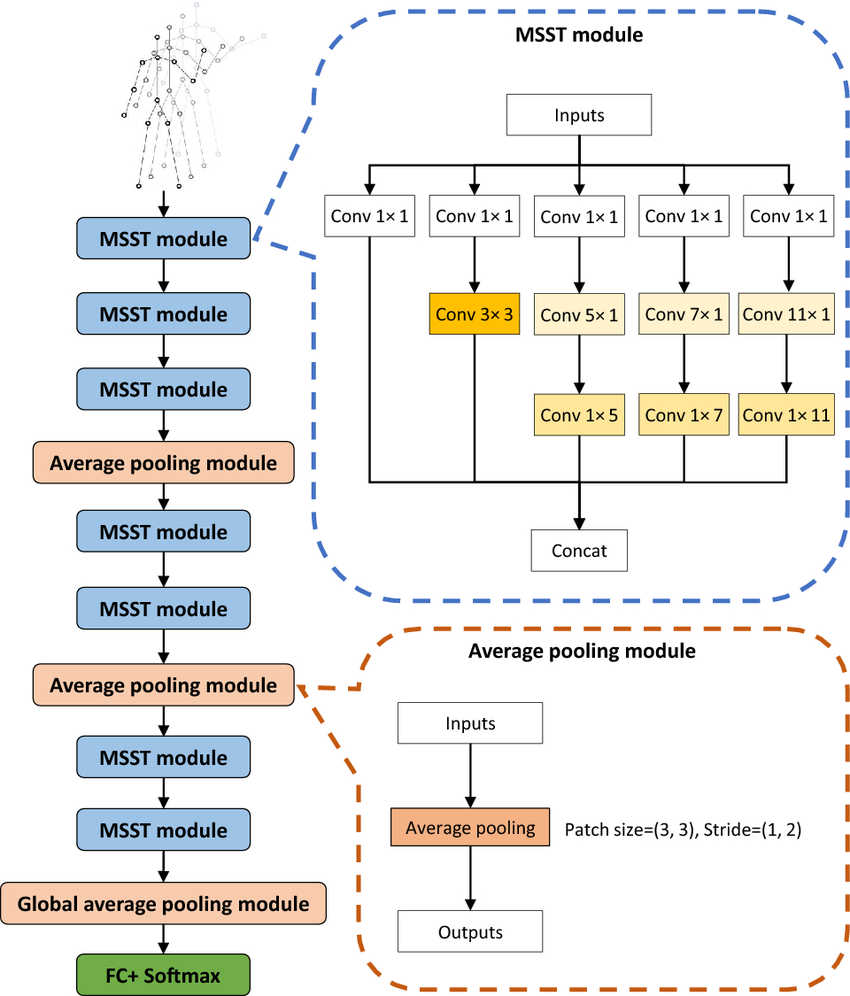


Figure 6: MSST module consists of five convolutional branches. The entire network is composed of multiple MSST modules and average pooling modules. Pooling reduces the spatial size of the feature map and suppresses overfitting

#### Training

##### Resnet

##### MSSTNET

Initially, a **joint-stream** dataset was extracted from video inputs using MediaPipe. From this dataset, we derived multiple data streams to capture various aspects of human motion. The **joint-motion stream** was computed by taking the difference in joint coordinates between consecutive frames, effectively capturing the temporal dynamics of joint movement. The **bone stream** was constructed to represent the skeletal structure, where bones are defined as the connections between adjacent joints. Subsequently, the **bone-motion stream** was obtained by calculating the temporal differences in bone vectors across successive frames, thereby modeling the dynamic behavior of the skeletal structure.

As a result, four distinct data representations of human skeletal motion were generated. Each of these streams was **independently** input into a separate MSSTNet model, enabling specialized feature extraction tailored to each motion representation. The outputs from the four MSSTNet models were then fused with the output from a ResNet model to enhance the overall prediction accuracy through multi-stream integration.

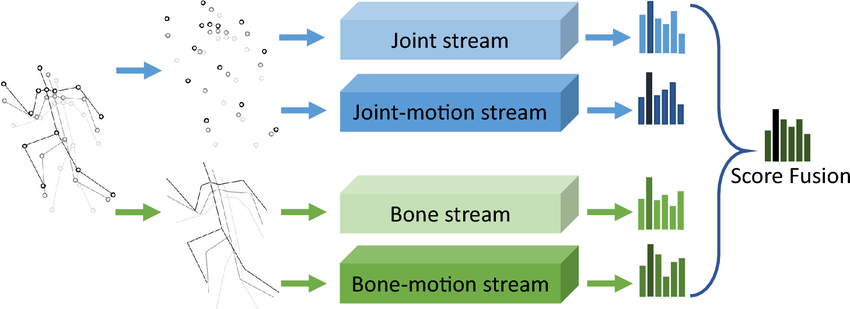


Figure 7: Input for MSSTNet model

The model was trained using the **Adam optimizer**, initialized with a learning rate of 0.01. Adam was selected for its adaptive learning rate mechanism, which offers improved convergence compared to conventional stochastic gradient descent methods. The **categorical cross-entropy loss function** was employed, reflecting the multi-class classification nature of the task, while **categorical accuracy** was used as the primary evaluation metric.

Where N is the number of action categories, YC is the one hot vector of true label, Y’ is the class probability scores.

Training was conducted over a maximum of 30 epochs with a batch size of 32, balancing computational efficiency and the stability of gradient updates. **The final model is chosen based on best val\_categorical\_accuracy.** To enhance generalization and mitigate overfitting, several advanced training strategies were applied. An early stopping mechanism with a patience value of seven epochs was utilized, halting the training process once validation accuracy ceased to improve and automatically restoring the best-performing model weights. Furthermore, the learning rate was adaptively adjusted using a plateau-based scheduling technique. If no improvement in validation accuracy was observed for four consecutive epochs, the learning rate was reduced by a factor of 0.5, with a minimum threshold set at 1e-5. This strategy enabled the model to escape local minima and promoted more refined convergence during the later stages of training.

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

AI-generated content may be incorrect.

Figure 8: Training and Validation Accuracy/Loss on the Joint Stream Data

A graph of a line and a line

AI-generated content may be incorrect.

Figure 9: Training and Validation Accuracy/Loss on the Joint-motion Stream Data

A graph of a line and a line

AI-generated content may be incorrect.

Figure 10: Training and Validation Accuracy/Loss on the Bone Stream Data

A graph of a graph

AI-generated content may be incorrect.

Figure 11: Training and Validation Accuracy/Loss on the Bone-motion Stream Data

# Result

We utilize the **MS COCO 2017 validation dataset**, which contains 5,000 samples. Due to CPU limitations, the dataset is divided into five subsets, each consisting of 1,000 samples. The results are then averaged over five trials.

The generated captions are evaluated using several metrics, including BLEU, METEOR, ROUGE-L, CIDEr, and SPICE. These metrics measure the similarity between the generated captions and the reference captions, providing a comprehensive assessment of the model's performance.

The results of the evaluation are as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **VGG16 - LSTM** | | **Transfomer - Pre** | **Transformer** |
| Bleu\_1 | 0.464 | 0.693 | | 0.593 |
| Bleu\_2 | 0.275 | 0.521 | | 0.403 |
| Bleu\_3 | 0.158 | 0.390 | | 0.266 |
| Bleu\_4 | 0.088 | 0.295 | | 0.178 |
| METEOR | 0.161 | 0.261 | | 0.189 |
| ROUGE\_L | 0.333 | 0.530 | | 0.438 |
| CIDEr | 0.473 | 0.998 | | 0.556 |
| SPICE | 0.127 | 0.190 | | 0.117 |

As shown in the table, the Transformer-Pre model consistently outperformed both the VGG16-LSTM and the Transformer models across all evaluation metrics. This superior performance can be attributed to the pre-trained Transformer architecture's ability to capture long-range dependencies within images and leverage knowledge from a large corpus of text and image data. Notably, the Transformer-Pre model achieved significantly higher scores in CIDEr, ROUGE-L, and BLEU-4 metrics, indicating its superior ability to generate captions that are more semantically accurate, grammatically correct, and closely aligned with the image content. While the VGG16-LSTM model also produced reasonable captions, particularly in BLEU-1 and BLEU-2 scores, its overall performance was inferior to the Transformer-Pre model.

These results highlight the advantages of the Transformer architecture for image captioning. The attention mechanism in the Transformer models allows it to capture long-range dependencies within the image and generate captions that are more contextually relevant. In contrast, the LSTM model, while effective for sequential tasks, may struggle to capture global information in images due to its recurrent nature.

Overall, the Transformer-Pre model demonstrated superior performance in generating image captions compared to the VGG16-LSTM model and the Transformer model. This highlights the potential of Transformer-based models for advancing the state-of-the-art in image captioning.

# Application

# Discussion and **Conclusion**

## Discussion

The results presented in the previous section indicate a clear performance advantage of the Transformer-Pre model over the other two models in the image captioning task. Notably, the Transformer-Pre model had a BLEU-1 score of 0.693, while the VGG16-LSTM and Transformer models had scores of 0.464 and 0.593, respectively. The Transformer-Pre model also had a higher CIDEr score of 0.998 compared to the VGG16-LSTM model's score of 0.473 and the Transformer model's score of 0.556. The Transformer-Pre model's superior performance across all metrics suggests that it is better able to capture the complexity of the images and generate more accurate and relevant captions.

This observation can be attributed to several factors, primarily the architectural differences between the models and the advantage of pre-training. The Transformer-Pre model, based on the Transformer architecture, leverages the self-attention mechanism, which allows it to capture long-range dependencies within the image and generate captions that are more contextually relevant. In contrast, the VGG16-LSTM model, while effective for sequential tasks, may struggle to capture global information in images due to its recurrent nature.

Furthermore, the Transformer-Pre model benefits from pre-training on a large corpus of text and image data. This pre-training allows the model to learn rich representations of language and visual concepts, which can be transferred to the image captioning task. Both the VGG16-LSTM and Transformer models are trained from scratch on the MS COCO dataset, which may limit their ability to generalize to unseen images.

**Comparison with State-of-the-Art Models**

While the Transformer-Pre model demonstrates strong performance on the MS COCO dataset, it is essential to place its results in the context of the broader field of image captioning. Several state-of-the-art models, such as mPLUG and OFA[18], have achieved even higher scores on the CIDEr metric, indicating their superior ability to generate captions that align with human consensus.

For example, mPLUG achieves the highest CIDEr score (1.551) among the models listed. Notably, mPLUG surpasses human performance on this metric, a significant achievement in the field. OFA follows closely with a CIDEr score of 1.549, demonstrating its effectiveness as a unified architecture for image captioning.

These models incorporate advanced techniques, such as cross-modal skip-connections and unified architectures, to achieve their impressive performance.[14]

The Transformer-Pre model, while not surpassing the absolute top performers, still demonstrates the potential of Transformer-based models for advancing the state-of-the-art in image captioning. Future work could explore incorporating techniques from other state-of-the-art models, such as cross-modal skip-connections or more advanced pre-training strategies, to further improve the performance of the Transformer-Pre model.

## Conclusion

In this report, we evaluated the performance of three image captioning models, VGG16-LSTM, Transformer and Transformer-Pre, on the MS COCO 2017 dataset. Our findings demonstrate that the Transformer-Pre model consistently outperforms the other two models across all evaluation metrics. This superior performance can be attributed to the Transformer architecture's ability to capture long-range dependencies within images and the benefits of pre-training on a large corpus of text and image data.

While the Transformer-Pre model shows promising results, it is essential to acknowledge that several state-of-the-art models achieve even higher performance on the MS COCO dataset. These models incorporate advanced techniques, such as cross-modal skip-connections and unified architectures, which contribute to their superior performance.

Future work could explore incorporating these techniques into the Transformer models to further improve their performance. Additionally, we will implement attention mechanism and bidirectional instead of using undirectional LSTM to enhance performance and address shortcomings of model. Besides, we can implement some another model architecture like VGG16 – BiGRU, Up – Down Attention Model which demonstrate one of SOTA current model.

Overall, this project highlights the potential of Transformer-based models for advancing the state-of-the-art in image captioning. As research in this field continues, we can expect to see even more sophisticated models that can generate highly accurate and human-like captions for images

# **References**

<https://www.qualcomm.com/developer/software/jester-dataset>

J. Materzynska, G. Berger, I. Bax and R. Memisevic, "The Jester Dataset: A Large-Scale Video Dataset of Human Gestures," 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea (South), 2019, pp. 2874-2882, doi: [10.1109/ICCVW.2019.00349.](https://doi.org/10.1109/ICCVW.2019.00349)

<https://ai.google.dev/edge/mediapipe/solutions/guide>

<https://github.com/google-ai-edge/mediapipe>

Cheng, Q., Cheng, J., Ren, Z. *et al.* Multi-scale spatial–temporal convolutional neural network for skeleton-based action recognition. *Pattern Anal Applic* 26, 1303–1315 (2023). [https://doi.org/10.1007/s10044-023-01156-w](https://link.springer.com/article/10.1007/s10044-023-01156-w#citeas)

Q. Cheng, J. Cheng, Z. Liu, Z. Ren and J. Liu, "A Dense-Sparse Complementary Network for Human Action Recognition based on RGB and Skeleton Modalities," *Expert Systems with Applications*, vol. 222, 2023, Art. no. 123061, doi: [10.1016/j.eswa.2023.123061](https://doi.org/10.1016/j.eswa.2023.123061).